AZ, BY, KG, KZ, MD, RU, TJ, TM), ヨーロッパ特許 (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI 特許 (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

規則4.17に規定する申立て:
— すべての指定国のための不利にならない開示又は新規性喪失の例外に関する申立て *(規則4.17(v))*

添付公開書類:
— 国際調査報告書
— 不利にならない開示又は新規性喪失の例外に関する申立て

２文字コード及び他の略語については、定期発行される各*PCT*ガゼットの巻頭に掲載されている「コードと略語のガイダンスノート」を参照。

明細書

疑似乱数発生方法及び疑似乱数発生器

[発明の属する技術分野]

本発明は、暗号通信やデジタル署名などで利用する疑似乱数を発生させる疑似乱数発生方法や、疑似乱数発生器、乱数発生プログラムに関する。

[従来の技術]

従来より、有線や無線により情報通信を行う際に、内容が第三者に漏れないように情報を暗号化して送信することが行われている。この暗号化方式の一つに逐次暗号方式（ストリーム暗号方式）がある。逐次暗号方式は、送信側と受信側で同一の疑似乱数を発生させ、送信側は疑似乱数のビット列と平文のビット列とを用いて暗号文のビット列を作成し暗号文として受信側に送出し、受信側は送信側から受信した暗号文のビット列と疑似乱数のビット列とを用いて平文のビット列を求め平文に復号化するものである。

第１６図は、従来の逐次暗号方式を説明する図である。送信側の暗号化装置１００は、疑似乱数発生器１０１と論理演算処理部１０２を有しており、受信側の復号化装置１１０は、疑似乱数発生器１１１と論理演算処理部１１２を有している。

暗号化装置１００の疑似乱数発生器１０１と復号化装置１１０の疑似乱数発生器１１１は、同一の秘密鍵を与えることによって互いに全く同一の疑似乱数を発生する論理構造を有している。また、暗号化装置１００の論理演算処理部１０２と復号化装置１１０の論理演算処理部１１２は、ビット単位で排他的論理和の演算処理を行う。

第１７図は、暗号化装置１００の疑似乱数発生器１０１を説明する図である。尚、復号化装置１１０の疑似乱数発生器１１１については、暗号化装置１００の疑似乱数発生器１０１と同一の構成を有するのでその詳細な説明を省略する。

疑似乱数発生器１０１は、非線形コンバイナ型の疑似乱数発生器（Nonlinear Combiner Generator）であり、第１７図に示すように、並列に配置した複数の線形フィードバックシフトレジスタ（Linear Feedback Shift Register;LFSR）１０

３と、非線形変換部１０４とを有しており、各線形フィードバックシフトレジスタ１０３から出力したビット列を非線形変換して疑似乱数を発生させる。本従来例では、各線形フィードバックシフトレジスタ１０３は、１回のシフト動作でそれぞれ１ビット（Ｘ１、Ｘ２、・・・X_L）を出力し、非線形変換部１０４は、各線形フィードバックシフトレジスタ１０３から入力されたビット列をもとに１ビットの疑似乱数を出力する構成を有している。

第１８図は、一般的な線形フィードバックシフトレジスタ１０３の構成を簡単に説明する図である。線形フィードバックシフトレジスタ１０３は、１ビットの情報を記憶できる複数のシフトレジスタ１０５と、複数の排他的論理和演算回路１０６とを有し、各シフトレジスタ１０５の出力と各排他的論理和演算回路１０６の一方の入力との間にはフィードバックタップ１０７が接続されている。フィードバックタップ１０７（c_{n-1}、c_{n-2}、・・・c_n）は、１のとき結線を示し、０のとき断線を示し、それぞれが予め１または０に定められている。

このシフトレジスタ１０５の個数をｎとすると、１つのシフトレジスタ１０５に注目したとき、出力系列の最大周期は、（２＾ｎ）－１となることが知られており、この系列をM系列という。（尚、「２＾ｎ」は、２のｎ乗（2^n）を意味する。以下、指数部分は、その前に「＾」を付して示す。）

例えば、第１４図に示す線形フィードバックシフトレジスタ１０３の場合、M系列を生成する特性多項式は、以下の式で表される。

$$C(x) = (X\hat{}n) + c_{n-1}(X\hat{}(n-1)) + \cdots + c_1X + 1$$

上記の特性多項式で第１項目の指数ｎは線形フィードバックシフトレジスタ１０３の次数、すなわちシフトレジスタの個数を示し、２項目以降の指数部分は、フィードバックタップによる結線位置を示している。上記の式に示す特性多項式が原始多項式となるようにすれば、線形フィードバックシフトレジスタは、M系列を出力する。

このような従来の非線形コンバイナ型疑似乱数発生器は、ビット単位の論理演算を基にした簡単なロジックで構成できるので、いわゆるハードウエアでの実装に適していると考えられている。

尚、従来より、線形フィードバックシフトレジスタからの出力を排他的論理和

等の演算処理によって変更することが提案されている（例えば、特許文献１参照。）。

【特許文献１】

特開平６－３４２２５７号

【発明が解決しようとする課題】

（第１の解決課題）

しかしながら、線形フィードバックシフトレジスタ１０３は、シフトレジスタ数の２倍の出力を観測することで、線形フィードバックシフトレジスタ１０３の構成、すなわちシフトレジスタ数及び結線位置と、初期値の全てを特定することが可能である。したがって、構成が固定された線形フィードバックシフトレジスタ１０３をそのまま疑似乱数発生器１０１に用いるには、暗号強度が弱く、安全性に問題がある。

また、線形フィードバックシフトレジスタ１０３は、特性多項式の変更によりシフトレジスタの結線位置や結線数を変更すると、線形フィードバックシフトレジスタの出力がM系列ではなく短周期となって暗号強度が低下するおそれがあることから、特性多項式は予めM系列を出力する値に固定されており、線形フィードバックシフトレジスタの構成を簡単に変更することはできないと考えられている。

（第２の解決課題）

また、従来の非線形コンバイナ型疑似乱数発生器は、線形フィードバックシフトレジスタ１０３で１ビット単位の演算を連続して繰り返し実行しなければならない。このような処理は、ハードウエアでは得意とするところであり、比較的高速に処理できるが、ソフトウエアでは苦手とするものであり、ハードウエアの場合と比較して処理速度が極端に遅くなる。

一方、非線形変換部１０４は、論理積や排他的論理和などの単純な演算を実行している。したがって、線形フィードバックシフトレジスタ１０３のスループットが、非線形変換部１０４のスループットよりも下回り、発生器全体の中で乱数ビット列を出力する部分、すなわち線形フィードバックシフトレジスタ１０３がボトルネックになってしまう。このため、従来の非線形コンバイナ型疑似乱数発生器は、ソフトウエアで実装した場合にはハードウエアで実装した場合よりも全

体のスループットが低下するという問題があり、ソフトウエアでは使用することが困難であった。

また、疑似乱数の暗号強度を十分に確保するためには、線形フィードバックシフトレジスタ１０３のシフトレジスタ１０５の数や、線形フィードバックシフトレジスタ１０３の個数をある程度の数以上必要とする。しかし、スループットは、線形フィードバックシフトレジスタ１０３のシフトレジスタ１０５の数が増加するほど、或いは線形フィードバックシフトレジスタ１０３の個数が増加するほど低くなるという、相反する関係にある。したがって、高い暗号強度を確保しつつ、高いスループットを実現することは困難であった。

本発明は、上述の第１及び第２の解決課題の少なくとも一方を解決すべくなされたものであり、その目的は、強い暗号強度を維持しつつ線形フィードバックシフトレジスタの構成を容易かつ動的に変更することができ、また、十分に高い暗号強度を確保しつつ、より高いスループットを実現できる疑似乱数発生方法等を提供することにある。

[課題を解決するための手段]

上記課題を解決する請求項１に記載の発明による疑似乱数発生方法は、ｎ個のシフトレジスタを有し、１周期分のビット数が（２＾ｎ）－１個となるビット列を出力可能な線形フィードバックシフトレジスタの初期値を設定する第１ステップと、所定の演算処理により初期値から線形フィードバックシフトレジスタの１周期分のビット数と互いに素である導出値を求める第２ステップと、導出値と線形フィードバックシフトレジスタの１周期分のビット数を２倍以上した値とを乗算して、第１線形フィードバックシフトレジスタにより出力させるビット列のビット数を算出する第３ステップと、その算出したビット数分のビット列を線形フィードバックシフトレジスタから初期値をもとに出力させる第４ステップと、その出力したビット列から導出値の間隔ごとにビット列を取り出して新ビット列を生成する第５ステップと、その新ビット列を出力可能な構成に線形フィードバックシフトレジスタの構成を再構成する第６ステップと、再構成した後の線形フィードバックシフトレジスタから初期値をもとに疑似乱数を発生させる第７ステップと、を有することを特徴とする。

この発明は、出力系列がM系列のビット列をｓ個ごとにサンプルしたビット列は、そのM系列の１周期分のビット数（＝（２＾ｎ）－１）と導出値が互いに素であるときには、他の構成を有する線形フィードバックシフトレジスタのM系列を構成し、また、少なくとも２周期分以上のビット数を有するビット列から線形フィードバックシフトレジスタを求めることができることを利用するものである。

この発明によると、ｎ個のシフトレジスタを有し、１周期分のビット数が（２＾ｎ）－１個となるビット列を出力可能な線形フィードバックシフトレジスタの初期値を設定し、所定の演算処理により初期値から線形フィードバックシフトレジスタの１周期分のビット数と互いに素である導出値を求める。

そして、導出値と線形フィードバックシフトレジスタの１周期分のビット数を２倍以上した値とを乗算して、第１線形フィードバックシフトレジスタにより出力させるビット列のビット数を算出し、その算出したビット数分のビット列を線形フィードバックシフトレジスタから初期値をもとに出力させ、その出力したビット列から導出値の間隔ごとにビット列を取り出して新ビット列を生成する。

そして、その新ビット列を出力可能な構成に線形フィードバックシフトレジスタの構成を再構成し、再構成した後の線形フィードバックシフトレジスタから初期値をもとに疑似乱数を発生させる。

これによれば、線形フィードバックシフトレジスタの構成を初期値に基づいて動的に変更することができ、変更後の線形フィードバックシフトレジスタからM系列のビット列を出力させることができる。したがって、解読者は、疑似乱数発生器から出力される疑似乱数に基づいて再構成前の線形フィードバックシフトレジスタの構成を得ることができず、初期値や秘密鍵も解読することができない。この結果、高い暗号強度を得ることができ、情報の秘匿性を保つことができる。

請求項２の発明は、請求項１に記載の疑似乱数発生方法において、初期値に対してハッシュ関数を施してハッシュ値を求め、そのハッシュ値に最も近似した素数を導出値として採用することを特徴とする。

この発明によると、初期値に対してハッシュ関数を施してハッシュ値を求め、そのハッシュ値に最も近似した素数を導出値として採用するので、導出値の推定困難度を高めることができ、より高度な秘匿性を得ることができる。

請求項３の発明は、請求項１または２に記載の疑似乱数発生方法において、線形フィードバックシフトレジスタの再構成は、バーレイキャンプマッセイアルゴリズムを用いて行われることを特徴とする。

この発明は、少なくとも２周期分以上のビット数を有するビット列から線形フィードバックシフトレジスタを求めることができるという、バーレイキャンプマッセイアルゴリズムを利用するものである。

請求項４の発明は、請求項１～３のいずれかに記載の疑似乱数発生方法において、第６ステップの発明は、発生させた疑似乱数を非線形変換する第７ステップを有することを特徴とする。この発明によると、発生させた疑似乱数を非線形変換するので、疑似乱数に非線形性を与えることができ、暗号強度を更に向上させることができる。

請求項５に記載の発明による疑似乱数発生器は、ｎ個のシフトレジスタを有し、１周期分のビット数が（２＾ｎ）－１個となるビット列を出力可能な線形フィードバックシフトレジスタと、秘密鍵に基づき線形フィードバックシフトレジスタの初期値を設定する初期値設定手段と、所定の演算処理により初期値から線形フィードバックシフトレジスタの１周期分のビット数と互いに素である導出値を求める導出値算出手段と、線形フィードバックシフトレジスタの１周期分のビット数を２倍以上した値と導出値とを乗算して、第１線形フィードバックシフトレジスタにより出力させるビット列のビット数を算出するビット数算出手段と、その算出したビット数分のビット列を線形フィードバックシフトレジスタから初期値をもとに出力させるビット列出力手段と、その出力したビット列から導出値の間隔ごとにビット列を取り出して新ビット列を生成する新ビット列生成手段と、その新ビット列を出力可能な構成に線形フィードバックシフトレジスタの構成を再構成する線形フィードバックシフトレジスタ再構成手段と、再構成後の線形フィードバックシフトレジスタから初期値をもとに疑似乱数を発生させる疑似乱数発生手段と、を有することを特徴とする。

この発明は、出力系列がM系列のビット列をｓ個ごとにサンプルしたビット列は、そのM系列の１周期分のビット数（＝（２＾ｎ）－１）と導出値ｓが互いに素であるときには、他の構成を有する線形フィードバックシフトレジスタのM系

列を構成し、また、少なくとも２周期分以上のビット数を有するビット列から線形フィードバックシフトレジスタを求めることができることを利用するものである。

この発明によると、ｎ個のシフトレジスタを有し、１周期分のビット数が（２＾ｎ）－１個となるビット列を出力可能な線形フィードバックシフトレジスタの初期値を設定し、所定の演算処理により初期値から線形フィードバックシフトレジスタの１周期分のビット数と互いに素である導出値を求める。

そして、導出値と線形フィードバックシフトレジスタの１周期分のビット数を２倍以上した値とを乗算して、第１線形フィードバックシフトレジスタにより出力させるビット列のビット数を算出し、その算出したビット数分のビット列を線形フィードバックシフトレジスタから初期値をもとに出力させ、その出力したビット列から導出値の間隔ごとにビット列を取り出して新ビット列を生成する。

そして、その新ビット列を出力可能な構成に線形フィードバックシフトレジスタの構成を再構成し、再構成した後の線形フィードバックシフトレジスタから初期値をもとに疑似乱数を発生させる。

これによれば、線形フィードバックシフトレジスタの構成を初期値に基づいて動的に変更することができ、変更後の線形フィードバックシフトレジスタからＭ系列のビット列を出力させることができる。したがって、解読者は、疑似乱数発生器から出力される疑似乱数に基づいて再構成前の線形フィードバックシフトレジスタの構成を得ることができず、初期値や秘密鍵も解読することができない。この結果、高い暗号強度を得ることができ、情報の秘匿性を保つことができる。

請求項６に記載の発明は、請求項５に記載の疑似乱数発生器において、線形フィードバックシフトレジスタ再構成手段の代わりに、新ビット列を出力可能な構成を有する第２の線形フィードバックシフトレジスタを生成する線形フィードバックシフトレジスタ生成手段を設け、疑似乱数発生手段は、第２の線形フィードバックシフトレジスタによって初期値をもとに疑似乱数を発生させることを特徴とする。この発明によると、線形フィードバックシフトレジスタを第１の線形フィードバックシフトレジスタと第２の線形フィードバックシフトレジスタの２つに分けることができ、より秘匿性の向上を図ることができる。

請求項７に記載の発明による疑似乱数発生器は、秘密鍵に基づいて所定ビット数を有する選択用乱数ビット列を出力する選択用乱数ビット列出力手段と、選択用乱数ビット列よりも大きなビット数を有する増幅乱数ビット列を予め複数格納した乱数テーブルと、選択用乱数ビット列出力手段から出力された選択用乱数ビット列を用いて乱数テーブルを参照することにより、乱数テーブル内の複数の増幅乱数ビット列の中から該当する増幅乱数ビット列を選択可能な増幅乱数ビット列選択手段と、増幅乱数ビット列選択手段により選択された増幅乱数ビット列を非線形関数によって非線形変換し疑似乱数として出力する非線形変換手段と、を有することを特徴とする。

この発明によると、秘密鍵に基づいて所定のビット数を有する選択用乱数ビット列を出力させ、その選択用乱数ビット列を用いて乱数テーブルを参照することにより、乱数テーブル内の複数の増幅乱数ビット列の中から該当する増幅乱数ビット列を選択し、非線形変換手段によって非線形変換して疑似乱数として出力させるので、小さなビット列を有する選択用乱数ビット列に基づいて、より大きなビット数を有する増幅乱数ビット列を得ることができる。

したがって、非線形変換手段に入力される乱数ビット列をより大きなビット数を有するものにすることができる。これにより、従来、ボトルネックとなっていた非線形変換手段よりも上流側の乱数ビット列を出力する部分のスループットを向上させ、非線形変換手段のスループットに近づけることができ、疑似乱数発生器全体のスループットを高速化することができる。

請求項８に記載の発明は、請求項７に記載の疑似乱数発生器において、秘密鍵が与えられることにより秘密鍵に基づいて増幅乱数ビット列を発生させ、乱数テーブルに格納して、乱数テーブルの初期設定を行う乱数テーブル初期設定手段を有することを特徴とする。

この発明によると、秘密鍵が与えられることにより秘密鍵に基づいて増幅乱数ビット列を発生させ、乱数テーブルに格納して、乱数テーブルの初期設定を行うので、秘密鍵を変更するごとに乱数テーブル内の初期値を変更することができる。したがって、暗号強度を増大させることができる。

請求項９に記載の発明は、請求項７または８に記載の疑似乱数発生器において、

選択用乱数ビット列出力手段が複数設けられ、乱数テーブルが各選択用乱数ビット列出力手段にそれぞれ対応するように設けられ、増幅乱数ビット列選択手段が各選択用乱数ビット列出力手段から各々出力された各選択用乱数ビット列を用いて各選択用乱数ビット列出力手段ごとに対応する乱数テーブルをそれぞれ参照し、各乱数テーブル内からそれぞれ該当する増幅乱数ビット列を選択し、非線形変換手段が各増幅乱数ビット列選択手段によって各乱数テーブルから選択された各増幅乱数ビット列を用いて非線形関数により非線形変換し疑似乱数として出力することを特徴とする。

この発明によると、各選択用乱数ビット列出力手段からそれぞれ選択用乱数ビット列が出力され、各選択用乱数ビット列を用いて各乱数テーブルがそれぞれ参照され、その参照により各乱数テーブルから選択された各増幅乱数ビット列を用いて非線形関数により非線形変換することによって疑似乱数を発生させるので、従来はボトルネックとなっていた乱数ビット列を出力する部分のスループットを向上させることができ、疑似乱数発生器全体のスループットを高速化することができる。

請求項１０に記載の発明は、請求項９に記載の疑似乱数発生器において、各選択用乱数ビット列出力手段ごとにそれぞれ複数の乱数テーブルを設け、増幅乱数ビット列選択手段により各乱数テーブル内から選択された各増幅乱数ビット列を選択用乱数ビット列出力手段ごとに排他的論理和演算して非線形変換手段に出力する排他的論理和演算処理手段を有することを特徴とする。

この発明によると、各乱数テーブルから選択された各増幅乱数ビット列が選択用乱数ビット列出力手段ごとに排他的論理和演算してから非線形変換手段に出力されるので、増幅乱数ビット列発生手段によって発生させた乱数ビット列をそのまま用いたものよりも、暗号強度を増大させることができる。

請求項１１に記載の発明は、請求項９または１０に記載の疑似乱数発生器において、所定のタイミングで各乱数テーブル同士の入れ替えを行う乱数テーブル入れ替え手段を有することを特徴とする。

この発明によると、所定のタイミングで各乱数テーブル同士の入れ替えを行うので、参照元となる乱数テーブルを変更することができる。したがって、固定さ

れたものよりも暗号強度を増大させることができる。

請求項１２に記載の発明は、請求項１１に記載の疑似乱数発生器において、乱数テーブル入れ替え手段が、各乱数テーブルを参照するために必要な選択用乱数ビット列を選択用乱数ビット列出力手段が出力するごとに、各乱数テーブル同士の入れ替えを行うことを特徴とする。

この発明は、上述の請求項に記載した所定のタイミングの一具体例を示したものである。これによると、各乱数テーブルを参照するために必要な選択用乱数ビット列を選択用乱数ビット列出力手段が出力するごとに各乱数テーブル同士の入れ替えを行う。したがって、短いサイクルで参照元となる乱数テーブルを変更することができ、暗号強度を更に増大させることができる。

請求項１３に記載の発明は、請求項１１または１２に記載の疑似乱数発生器において、乱数テーブル入れ替え手段は、各乱数テーブルの個数と等しい個数の乱数テーブル入れ替え用乱数を発生させ、乱数テーブル入れ替え用乱数を乱数テーブルのテーブル番号として各乱数テーブルに付与し、テーブル番号をもとに予め設定された規則に従って乱数テーブルの順番を入れ替えることを特徴とする。

この発明は、上述の乱数テーブル入れ替え手段についての具体的な一例を示したものである。これによると、乱数テーブル入れ替え用乱数を発生させ、乱数テーブルのテーブル番号として各乱数テーブルに付与し、その付与したテーブル番号をもとに乱数テーブルの順番を入れ替える。したがって、乱数テーブルの順番を簡単かつ迅速に入れ替えることができ、非線形変換手段よりも上流側のスループットを向上させ、非線形変換手段のスループットに近づけることができ、疑似乱数発生器全体のスループットを高速化することができる。

請求項１４に記載の発明は、コンピュータを以下の手段として機能させるための疑似乱数発生プログラムであり、秘密鍵に基づいて所定ビット数を有する選択用乱数ビット列を出力させる選択用乱数ビット列出力手段と、選択用乱数ビット列よりも大きなビット数を有する増幅乱数ビット列を予め複数格納した乱数テーブルと、選択用乱数ビット列出力手段から出力された選択用乱数ビット列を用いて前記乱数テーブルを参照することにより、前記乱数テーブル内の複数の増幅乱数ビット列の中から該当する増幅乱数ビット列を選択可能な増幅乱数ビット列選

択手段と、増幅乱数ビット列選択手段により選択された増幅乱数ビット列を非線形関数によって非線形変換し疑似乱数として出力する非線形変換手段として機能させるためのプログラムである。

この発明によると、秘密鍵に基づいて所定のビット数を有する選択用乱数ビット列を出力させ、その選択用乱数ビット列を用いて乱数テーブルを参照することにより、乱数テーブル内の複数の増幅乱数ビット列の中から該当する増幅乱数ビット列を選択し、非線形変換手段によって非線形変換して疑似乱数として出力させるので、小さなビット列を有する選択用乱数ビット列に基づいて、より大きなビット数を有する増幅乱数ビット列を得ることができる。

したがって、非線形変換手段に入力される乱数ビット列をより大きなビット数を有するものにすることができる。これにより、従来は、ボトルネックとなっていた非線形変換手段よりも上流側の乱数ビット列を出力する部分のスループットを向上させ、非線形変換部のスループットに近づけることができ、疑似乱数発生器全体のスループットを高速化することができる。

請求項１５に記載の発明は、請求項１４に記載の疑似乱数発生プログラムにおいて、秘密鍵が与えられることにより秘密鍵に基づいて増幅乱数ビット列を発生させ、乱数テーブルに格納して、乱数テーブルの初期設定を行う乱数テーブル初期設定手段としてコンピュータを機能させることを特徴とする。

この発明によると、秘密鍵が与えられることにより秘密鍵に基づいて増幅乱数ビット列を発生させ、乱数テーブルに格納して、乱数テーブルの初期設定を行うので、秘密鍵を変更するごとに乱数テーブル内の初期値を変更することができる。したがって、暗号強度を増大させることができる。

請求項１６に記載の発明は、請求項１４または１５に記載の疑似乱数発生プログラムにおいて、選択用乱数ビット列出力手段は、複数設けられ、乱数テーブルは、各選択用乱数ビット列出力手段にそれぞれ対応するように設けられ、増幅乱数ビット列選択手段は、各選択用乱数ビット列出力手段から各々出力された各選択用乱数ビット列を用いて各選択用乱数ビット列出力手段ごとに対応する乱数テーブルをそれぞれ参照し、各乱数テーブル内からそれぞれ該当する増幅乱数ビット列を選択し、非線形変換手段は、各増幅乱数ビット列選択手段によって各乱数

テーブルから選択された各増幅乱数ビット列を用いて非線形関数により非線形変換し疑似乱数として出力することを特徴とする。

この発明によると、各選択用乱数ビット列出力手段からそれぞれ選択用乱数ビット列が出力され、各選択用乱数ビット列を用いて各乱数テーブルがそれぞれ参照され、その参照により各乱数テーブルから選択された各増幅乱数ビット列を用いて非線形関数により非線形変換することによって疑似乱数を発生させるので、従来はボトルネックとなっていた非線形変換手段よりも上流側の乱数ビット列を出力する部分のスループットを向上させ、非線形変換手段のスループットに近づけることができ、疑似乱数発生器全体のスループットを高速化することができる。

請求項１７に記載の発明は、請求項１６に記載の疑似乱数発生プログラムにおいて、各選択用乱数ビット列出力手段ごとにそれぞれ複数の乱数テーブルを設け、増幅乱数ビット列選択手段により各乱数テーブル内から選択された各増幅乱数ビット列を選択用乱数ビット列出力手段ごとに排他的論理和演算して非線形変換手段に出力する排他的論理和演算処理手段としてコンピュータを機能させることを特徴とする。

この発明によると、各乱数テーブルから選択された各増幅乱数ビット列が選択用乱数ビット列出力手段ごとに排他的論理和演算してから非線形変換手段に出力されるので、増幅乱数ビット列発生手段によって発生させた乱数ビット列をそのまま用いたものよりも、暗号強度を増大させることができる。

請求項１８に記載の発明は、請求項１６または１７に記載の疑似乱数発生プログラムにおいて、所定のタイミングで前記各乱数テーブル同士の入れ替えを行う乱数テーブル入れ替え手段としてコンピュータを機能させることを特徴とする。

この発明によると、所定のタイミングで各乱数テーブル同士の入れ替えを行うので、参照元となる乱数テーブルを変更することができる。したがって、固定されたものよりも暗号強度を増大させることができる。

請求項１９に記載の発明は、請求項１８に記載の疑似乱数発生プログラムにおいて、乱数テーブル入れ替え手段は、各乱数テーブルを参照するために必要な選択用乱数ビット列を選択用乱数ビット列出力手段が出力するごとに、各乱数テーブル同士の入れ替えを行うことを特徴とする。

この発明は、上述の請求項に記載した所定のタイミングの一具体例を示したものである。これによると、各乱数テーブルを参照するために必要な選択用乱数ビット列を選択用乱数ビット列出力手段が出力するごとに各乱数テーブル同士の入れ替えを行う。したがって、短いサイクルで参照元となる乱数テーブルを変更することができ、暗号強度を更に増大させることができる。

請求項２０に記載の発明は、請求項１８または１９に記載の疑似乱数発生プログラムにおいて、乱数テーブル入れ替え手段は、各乱数テーブルの個数と等しい個数の乱数テーブル入れ替え用乱数を発生させ、乱数テーブル入れ替え用乱数を乱数テーブルのテーブル番号として各乱数テーブルに付与し、テーブル番号をもとに予め設定された規則に従って乱数テーブルの順番を入れ替えることを特徴とする。

この発明は、上述の乱数テーブル入れ替え手段についての具体的な一例を示したものである。これによると、乱数テーブル入れ替え用乱数を発生させ、乱数テーブルのテーブル番号として各乱数テーブルに付与し、その付与したテーブル番号をもとに乱数テーブルの順番を入れ替える。したがって、乱数テーブルの順番を簡単かつ迅速に入れ替えることができ、非線形変換手段よりも上流側のスループットを向上させ、非線形変換手段のスループットに近づけることができ、疑似乱数発生器全体のスループットを高速化することができる。

[発明の実施の形態]

（第１の実施の形態）

次に、本発明の第１の実施の形態について図に基づいて説明する。

第１図は、本実施の形態における疑似乱数発生器１を説明する図である。本実施の形態では、非線形コンバイナ型の疑似乱数発生器１を例に説明する。

疑似乱数発生器１は、利用者から与えられる秘密鍵に基づいて初期値を設定する初期値設定部（図示せず）と、初期値設定部から受け取った初期値をもとに疑似乱数を生成する複数の疑似乱数生成部１０と、これら複数の疑似乱数生成部１０の出力側に各々接続され、各疑似乱数生成部１０から出力される疑似乱数を非線形変換する非線形変換部２０を有している。

初期値設定部は、利用者から与えられる秘密鍵をビット列に変換し、疑似乱数

生成部１０の数に分割して、後述する疑似乱数生成部１０の線形フィードバックシフトレジスタ１１にそれぞれ割り当てる初期値を生成する処理を行う。

疑似乱数生成部１０は、Ｌ個が互いに並列に配置されており、それぞれ線形フィードバックシフトレジスタ１１と、線形フィードバックシフトレジスタ再構成手段１２を有している。

線形フィードバックシフトレジスタ１１は、従来技術で説明したものと同様に、１ビットの情報を記憶できるｎ個のシフトレジスタと、排他的論理和演算回路を有している。そして、本実施の形態では、１周期分のビット数ｍが（２＾ｎ）－１個となるビット列、いわゆるM系列を出力可能な構成に予め設定されている。

第２図は、本実施の形態における線形フィードバックシフトレジスタ１１の初期多項式を例示するものである。初期多項式は、M系列を出力するように予め設定されている特性多項式であり、１項目の指数部（第２図では「＾」で表している）がシフトレジスタの個数を示し、２項目以降の指数部が排他的論理和演算回路に接続された結線位置を示している。例えば、１段目の線形フィードバックシフトレジスタ１１（ＬＦＳＲ１）は、１３１個のシフトレジスタを有し、８番目、３番目、２番目のシフトレジスタがフィードバックタップによって排他的論理和演算回路に接続されていることを示している。尚、本実施の形態では、シフトレジスタの個数ｎは、全て素数個に設定されている。

線形フィードバックシフトレジスタ再構成手段１２は、線形フィードバックシフトレジスタ１１の構成を秘密鍵によって動的に変更して再構成するものである。具体的には、出力系列がM系列のビット列をｓ個ごとにサンプルした新ビット列は、M系列の１周期分のビット数ｍ（＝（２＾ｎ）－１）と導出値ｓとが互いに素であるとき、すなわち、１以外の共通の約数を持たないときは、他の構成を有する線形フィードバックシフトレジスタのM系列になり、また、バーレイキャンプマッセイアルゴリズムによって、少なくとも２周期分以上のビット数を有するビット列から、そのビット列を出力可能な等価で最小の線形フィードバックシフトレジスタの特性多項式を求めることができることを利用して、線形フィードバックシフトレジスタ１１の再構成を行う。

線形フィードバックシフトレジスタ再構成手段１２は、初期値設定部によって

与えられた初期値から導出値ｓを算出し、導出値ｓと線形フィードバックシフトレジスタ１１の１周期分のビット数ｍ（＝（２＾ｎ）－１）を２倍した値２ｍとを乗算し、線形フィードバックシフトレジスタ１１から出力させるビット列のビット数２ｍｓを算出する。

そして、初期値をもとに線形フィードバックシフトレジスタ１１から２ｍｓ個のビット列を出力させ、その２ｍｓ個のビット列から導出値ｓの間隔ごとにビット列を取り出して新ビット列を生成し、その新ビット列を用いてバーレイキャンプマッセイアルゴリズムにより線形フィードバックシフトレジスタ１１の構成を変更する。

尚、本実施の形態では、線形フィードバックシフトレジスタ１１から出力させるビット列のビット数が２ｍｓ個である場合を例に説明しているが、新ビット列のビット数が2m個以上であれば、等価な最小の線形フィードバックシフトレジスタを求めることができるので、２ｍｓ個以上であればよい。

バーレイキャンプマッセイアルゴリズムとは、線形フィードバックシフトレジスタ１１のシフトレジスタの個数ｎ（線形複雑度）の２倍以上のビット数を有するビット列を入手することで、そのビット列を出力可能な等価な最小の線形フィードバックシフトレジスタを得ることができるというアルゴリズムである。バーレイキャンプマッセイアルゴリズムについては、例えば、文献１「暗号理論入門（第２版）」、共立出版社、岡本栄司著、２００２年４月１０日発行、に詳細に説明されている。

次に、上記構成を有する疑似乱数発生器１の動作について第３図のフローチャートを用いて以下に説明する。

まず最初に、初期値設定部によって初期値が設定される（ステップＳ１）。初期値は、利用者から与えられる秘密鍵を所定の演算処理によって分割することによって設定される。

例えば、秘密鍵の長さが１６バイトで「ＡＢＣＤＥＦＧＨＩＪＫＬＭＮＯＰ」であり、疑似乱数生成部１０が８段の場合には、初期値は下記のように設定される。

ＬＦＳＲ１　ＡＢ＋Ｘ’ＦＦ’埋め込み文字（Ｐａｄｄｉｎｇ）

ＬＦＳＲ２　ＣＤ＋Ｘ’ＦＦ’埋め込み文字（Ｐａｄｄｉｎｇ）
ＬＦＳＲ３　ＥＦ＋Ｘ’ＦＦ’埋め込み文字（Ｐａｄｄｉｎｇ）
ＬＦＳＲ４　ＧＨ＋Ｘ’ＦＦ’埋め込み文字（Ｐａｄｄｉｎｇ）
ＬＦＳＲ５　ＩＪ＋Ｘ’ＦＦ’埋め込み文字（Ｐａｄｄｉｎｇ）
ＬＦＳＲ６　ＫＬ＋Ｘ’ＦＦ’埋め込み文字（Ｐａｄｄｉｎｇ）
ＬＦＳＲ７　ＭＮ＋Ｘ’ＦＦ’埋め込み文字（Ｐａｄｄｉｎｇ）
ＬＦＳＲ８　ＯＰ＋Ｘ’ＦＦ’埋め込み文字（Ｐａｄｄｉｎｇ）

ここでは、初期値は、秘密鍵「ＡＢＣＤＥＦＧＨＩＪＫＬＭＮＯＰ」を、「ＡＢ」、「ＣＤ」、・・・、「ＯＰ」の２文字ずつに分割し、残りのシフトレジスタを埋め込み文字（Ｐａｄｄｉｎｇ）で埋めることによって設定される。尚、上述の初期値設定方法は、実施例の一つであり、他の方法によって設定してもよい。

初期値設定部において秘密鍵から初期値が設定されると、各初期値は、各疑似乱数生成部１０にそれぞれ入力され、線形フィードバックシフトレジスタ１１のシフトレジスタ内にセットされる。

次に、線形フィードバックシフトレジスタ再構成手段１２によって、線形フィードバックシフトレジスタ１１の構成を再構成する処理が行われる（ステップＳ２～ステップＳ６）。

ここでは、まず、所定の演算処理により初期値から線形フィードバックシフトレジスタ１１の１周期分のビット数ｍと互いに素である導出値ｓを算出する（ステップＳ２）。導出値ｓは、初期値に対して、例えばＭＤ５（Message Digest 5）などのハッシュ関数を施してハッシュ値を求め、そのハッシュ値に最も近似した素数が採用される。したがって、導出値の推定困難度を高めることができ、より高度な秘匿性を得ることができる。尚、導出値ｓは、初期値から求めることができ、かつビット数ｍと互いに素であればよく、上記の算出方法によって求められるものに限定されない。但し、秘匿性を維持するために、上記所定の演算処理は、一方向性を満足しうる演算処理でなければならない。

導出値ｓを算出すると、次に、線形フィードバックシフトレジスタ１１から出力させるビット列のビット数２ｍｓを算出する（ステップＳ３）。線形フィードバックシフトレジスタ１１から出力させるビット列のビット数２ｍｓは、線形フ

ィードバックシフトレジスタ１１の１周期分のビット数m（＝（２＾ｎ）－１）を２倍した値と、導出値ｓとを乗算することによって求められる。

そして次に、線形フィードバックシフトレジスタ１１から初期値をもとに２ｍｓ個のビット数を有するビット列を出力させ（ステップＳ４）、そのビット列から新ビット列を生成する（ステップＳ５）。新ビット列は、２ｍｓ個のビット列から導出値ｓの間隔ごとに取り出したビット列によって構成され、そのビット数は２ｍ個となる。

ここで、出力系列がM系列のビット列をｓ個ごとにサンプルしたビット列は、そのM系列の１周期分のビット数ｍと導出値ｓとが互いに素であれば、他の構成を有する線形フィードバックシフトレジスタのM系列となることから、この新ビット列も、M系列となる。

そして、その新ビット列に基づいて線形フィードバックシフトレジスタ１１の構成を再構成する（ステップＳ６）。線形フィードバックシフトレジスタ１１の再構成は、バーレイキャンプマッセイアルゴリズムを用いて行われる。バーレイキャンプマッセイアルゴリズムによれば、少なくとも２周期分以上のビット数を有するビット列があれば、かかるビット列を出力可能な等価で最小の線形フィードバックシフトレジスタを求めることができるので、２ｍ個のビット数を有する新ビット列から新たな線形フィードバックシフトレジスタの特性多項式を導出して、再構成を行う。

再構成後の線形フィードバックシフトレジスタ１１は、再構成前と同一の次数及び異なる結線の特性多項式を有し、同一の初期値を与えた場合に、再構成前と異なるM系列を出力可能な構成を有する。

線形フィードバックシフトレジスタ再構成手段１２による線形フィードバックシフトレジスタ１１の再構成が終了すると、再構成された線形フィードバックシフトレジスタ１１から初期値をもとに疑似乱数を発生させる処理が行われる（ステップＳ７）。これにより、疑似乱数生成部１０から再構成前とは異なるM系列の疑似乱数が発生される。

各疑似乱数生成部１０から出力された疑似乱数は、それぞれ非線形変換部２０に入力され、非線形変換部２０で所定の非線形関数ｆ（ｘ）に基づいて非線形変

換される（ステップＳ８）。これにより、疑似乱数に非線形性を与えることができ、暗号強度を更に向上させることができる。

上記構成を有する疑似乱数発生器１によれば、線形フィードバックシフトレジスタ１１の構成を初期値に基づいて容易かつ動的に変更することができ、変更後もM系列を出力させることができる。したがって、解読者は、再構成前の線形フィードバックシフトレジスタの構成を取得することができない。これにより、従来、線形フィードバックシフトレジスタの構成が既知であることを前提に成り立っていた既存の暗号解読法は、成立しなくなる。したがって、高い暗号強度を得ることができ、情報の秘匿性を保つことができる。

尚、上述の実施の形態では、非線形コンバイナ型の疑似乱数発生器１を例に説明したが、非線形コンバイナ型に限定されるものではなく、線形フィードバックシフトレジスタを用いる疑似乱数発生器であればよく、例えばブロック型暗号方式に用いられる疑似乱数発生器に用いてもよい。

また、上記のステップＳ６で、新ビット列に基づいて線形フィードバックシフトレジスタ１１の構成を再構成する代わりに、新ビット列を出力可能な構成を有する第２の線形フィードバックシフトレジスタを生成し、ステップＳ７で、その第２の線形フィードバックシフトレジスタによって初期値をもとに疑似乱数を発生させてもよい。これによれば、線形フィードバックシフトレジスタを２つに分けることができ、より秘匿性の向上を図ることができる。また、第１の実施の形態における疑似乱数発生器１は、ソフトウエアやハードウエアのいずれによって構成してもよい。

（第２の実施の形態）

次に、本発明の第２の実施の形態について図に基づいて説明する。

第４図は、第２の実施の形態における疑似乱数発生器１の機能を概略的に示す説明図である。本実施の形態における疑似乱数発生器１は、疑似乱数発生プログラムをコンピュータハードウエア上で実行することによって実現される非線形コンバイナ型の疑似乱数発生器１である。尚、本実施の形態では、暗号化装置（従来技術を参照）に組み込まれた場合のものを例に説明し、復号化装置のものについては同様であるのでその詳細な説明を省略する。

疑似乱数発生器１は、第４図に示すように、乱数ビット列出力部５０と、乱数ビット列増幅部６０と、非線形変換部８０を有している。乱数ビット列出力部５０は、α個の選択用乱数ビット列出力手段５１を備えている。選択用乱数ビット列出力手段５１_{1}～５１_{α}は、利用者から与えられるＬｋビットの秘密鍵ＫをもとにＮｉビットを有する選択用乱数ビット列を連続して出力するものであり、例えば線形フィードバックシフトレジスタによって構成される。

乱数ビット列増幅部６０は、Ｎｉビットの選択用乱数ビット列を与えることによりＮｉビットよりも大きなビット数であるＮｏビットの増幅乱数ビット列を出力するように構成されており、乱数テーブル部６１と排他的論理和演算処理手段６３を備えている。

乱数テーブル部６１は、（２＾Ｎｉ）個の乱数ビット列を格納したα×β（以下、単に「αβ」と記す）個の乱数テーブル６２によって構成されている。そして、第４図に示すように、各選択用乱数ビット列出力手段５１ごとにβ（複数）個の乱数テーブル６２が対応するように設けられている。第５図は、一の乱数テーブルの構成を説明する概略図である。各乱数テーブル６２は、第５図に示すように、（２＾Ｎｉ）個でかつ０～（２＾Ｎｉ）－１のインデックス番号が付与されたインデックス部Ｒｉと、各インデックス番号に一対一で対応して設けられ上述の増幅乱数ビット列を格納可能なビット列格納部Ｒｏを有している。

そして、乱数ビット列出力部５０の選択用乱数ビット列出力手段５１から出力された選択用乱数ビット列を引数として、該当するインデックス部Ｒｉのインデックス番号を選択し、そのインデックス番号に対応する乱数ビット列格納部ＲｏからＮｏビットの増幅乱数ビット列を選択できるように構成されている。

排他的論理和演算処理手段６３は、乱数テーブル６２_{1}～$６２_{\alpha\beta}$の参照によって抽出されたαβ個の増幅乱数ビット列を各選択用乱数ビット列出力手段５１ごとに排他的論理和演算処理し、α個の増幅乱数ビット列とし、非線形変換部８０に出力するように構成されている。これにより、乱数テーブル６２_{1}～$６２_{\alpha\beta}$から読み出した増幅乱数ビット列をそのまま非線形変換部８０に出力するのではなく、暗号強度が増幅乱数ビット列そのものに依存することを防止し、暗号強度を更に向上させている。

第６図は、乱数ビット列増幅部６０内に構成される各構成要素を説明する概念図である。上述の乱数ビット列増幅部６０は、第６図に示すように、その内部機構として増幅乱数ビット列選択手段６４を備えている。増幅乱数ビット列選択手段６４は、各選択用乱数ビット列出力手段５１_{1}～５１_{α}から出力された選択用乱数ビット列を引数として各乱数テーブル６２_{1}～$６２_{\alpha\beta}$をそれぞれ参照し、引数と等しい値を有するインデックス番号に対応するビット列格納部Ｒｏから増幅乱数ビット列をそれぞれ選択するように構成されている。

また、乱数ビット列増幅部６０は、乱数テーブル部６１の初期設定を行う乱数テーブル初期設定手段６５と、その乱数テーブル初期設定手段６５により乱数テーブル部６１内に設定される増幅乱数ビット列を発生させる増幅乱数ビット列発生手段６６を備えている。

乱数テーブル初期設定手段６５は、増幅乱数ビット列発生手段６６によって発生させた乱数ビット列をＮｏビットごとに分割して各乱数テーブル６２_{1}～$６２_{\alpha\beta}$の全ての乱数ビット列格納部Ｒｏに格納する処理を行うものであり、本実施の形態では、第１番目の選択用乱数ビット列出力手段５１_{1}に対応する乱数テーブル６２_{1}から第α番目の選択用乱数ビット列出力手段５１_{α}に対応する乱数テーブル$６２_{\alpha\beta}$まで順番に格納するように構成されている。

増幅乱数ビット列発生手段６６は、秘密鍵Ｋをもとに乱数ビット列を出力するものであり、本実施の形態では、ＲＣ４　Ｓｙｐｐｅｔｒｉｃ　Ｓｔｒｅａｐ　Ｃｉｐｈｅｒ（ＲＳＡ　Ｄａｔａ　Ｓｅｃｕｒｉｔｙ　Ｉｎｃ．製）を用いている。しかし、通常の線形フィードバックシフトレジスタなどの疑似乱数ビット列を高速に出力できるもの（主としてストリーム型暗号）であれば他のものであってもよい。

また、第６図に示すように、乱数ビット列増幅部６０は、所定のタイミングで乱数テーブル６２_{1}～$６２_{\alpha\beta}$の順番を入れ替える処理を行う乱数テーブル入れ替え手段６７と、その乱数テーブル入れ替え手段６７が乱数テーブルの順番入れ替え処理を行うために使用する順番入れ替え用乱数を発生させる入れ替え用乱数発生手段６８を備えている。

乱数テーブル入れ替え手段６７は、入れ替え用乱数発生手段６８により発生さ

せた入れ替え用乱数を、乱数テーブルのテーブル番号として、その発生させた順番で各乱数テーブル62_1～$62_{\alpha\beta}$に順次付与し、その付与した乱数をもとに乱数テーブルの順番を入れ替える処理を行い、乱数テーブル部６１内の増幅乱数ビット列の順番をテーブル単位で変更する。

入れ替え用乱数発生手段６８は、任意の秘密鍵Ｋ０をもとに乱数テーブル入れ替え用乱数を発生させる処理を行うものであり、乱数ビット列出力部５０からＮｉビットを有するα個の選択用乱数ビット列を入力するごとに、αβ個の入れ替え用乱数を発生するように構成されている。任意の秘密鍵Ｋ０は、本実施の形態では、上述の増幅乱数ビット列発生手段６６に秘密鍵Ｋを与えて出力させた増幅乱数ビット列からＬｋビット分だけ取り出した値を用いている。しかし、これに拘束されるものではなく、例えば、他の手段によって発生させたり、別途にユーザに入力させてもよい。

非線形変換部８０は、α入力１出力の１次無相関な非線形関数ｆ（ｘ）を有しており、乱数ビット列増幅部６０から出力されたα個の増幅乱数ビット列を非線形変換し、Ｎｏビットを有する１個の乱数ビット列を疑似乱数Ｚとして出力するように構成されている。

尚、秘密鍵Ｋは、１２８ビット、２５６ビット、５１２ビット、１０２４ビットの中から選択され、また、選択用乱数ビット列出力手段５１の数α、各選択用乱数ビット列出力手段５１に対応する乱数テーブルの数β、及び選択用乱数ビット列のビット数Ｎｉは、互いにかけ算した値が秘密鍵Ｋのビット数Ｌｋに等しいという条件の範囲内で選択される。

次に、疑似乱数発生方法について第７図に基づき説明する。第７図は、本実施の形態における疑似乱数発生方法を説明するフローチャートである。

まず最初に、乱数ビット列出力部５０は、ユーザからＬｋビットを有する任意の秘密鍵Ｋの入力を受けると（ステップＳ１１）、その秘密鍵Ｋを用いて選択用乱数ビット列出力手段５１の初期値を設定する（ステップＳ１２）。例えば、選択用乱数ビット列出力手段５１が線形フィードバックシフトレジスタによって構成されている場合には、その秘密鍵Ｋに基づいて各シフトレジスタ内に格納される初期値の設定が行われる。

各選択用乱数ビット列出力手段５１の初期値を設定すると、次に、乱数テーブル初期設定手段６５により乱数テーブル部６１の初期設定が行われる（ステップＳ１３）。ここでは、まず、増幅乱数ビット列発生手段６６に秘密鍵Ｋが与えられ、高速で乱数ビット列が発生される。この増幅乱数ビット列発生手段２４により発生された乱数ビット列は、乱数テーブル初期設定手段６５によって、Ｎｏビットごとに分割され、増幅乱数ビット列として各乱数テーブル６２_{1}～$６２_{\alpha\beta}$の全ての乱数ビット列格納部Ｒｏに順次格納される。このように、秘密鍵Ｋが与えられることによって、乱数テーブル部６１の初期設定が予め行われる。

上述のステップＳ１１～ステップＳ１３により選択用乱数ビット列出力手段５１と乱数テーブル部６１の初期値の設定が行われると、待機状態となる。そして、平文の暗号化装置（従来技術を参照）への入力をトリガとして、乱数ビット列の増幅処理が開始される（ステップＳ１４～Ｓ１６）。まず最初に、各選択用乱数ビット列出力手段５１によって、それぞれＮｉビットを有する選択用乱数ビット列を乱数テーブルの数であるβ個出力させ、乱数ビット列増幅部６０内に記憶させる（ステップＳ１４）。

それから、乱数テーブル順番入れ替え手段２６により乱数テーブル６２_{1}～$６２_{\alpha\beta}$の順番入れ替え処理を行う（ステップＳ１５）。ここでは、まず、入れ替え用乱数発生手段６８によりαβ個の入れ替え用乱数を発生させ、乱数テーブルの順番入れ替え用のテーブル番号として、各乱数テーブル６２_{1}～$６２_{\alpha\beta}$に付与する。これらのテーブル番号は、その発生の順番で乱数テーブル６２_{1}から順次付与される。

したがって、各乱数テーブル６２_{1}～$６２_{\alpha\beta}$には、１～αβまでのテーブル番号が順不同に付与される。そして、その付与したテーブル番号をもとに乱数テーブル部６１内の増幅乱数ビット列の順番を各乱数テーブル単位で入れ替える処理が行われる。これにより、乱数テーブル部６１の乱数ビット列格納部Ｒｏに格納されている増幅乱数ビット列は、昇順や降順などの予め設定した規則に従って、各乱数テーブル単位で入れ替えられる。

乱数テーブル６２_{1}～$６２_{\alpha\beta}$の順番入れ替える処理が終了すると、増幅乱数ビット列選択手段６４により、各乱数テーブル６２_{1}～$６２_{\alpha\beta}$内から該当する増幅

乱数ビット列を選択する増幅乱数ビット列選択処理が行われる（ステップＳ１６）。増幅乱数ビット列選択手段６４は、乱数ビット増幅部２０に格納されている各選択用乱数ビット列を用いて、対応する乱数テーブル６２_{1}～$６２_{\alpha\beta}$をそれぞれ参照し、各乱数テーブル６２_{1}～$６２_{\alpha\beta}$内からそれぞれ該当する増幅乱数ビット列を選択する。

増幅乱数ビット列の選択処理が終了すると、次に、排他的論理和演算処理手段６３により排他的論理和演算処理が行われる（ステップＳ１７）。排他的論理和演算処理手段６３は、各乱数テーブル６２_{1}～$６２_{\alpha\beta}$から読み出したαβ個の増幅乱数ビット列を、各選択用乱数ビット列出力手段５１単位で排他的論理和演算処理する。これにより、Ｎｏビットを有するα個の新たな増幅乱数ビット列が生成される。

そして、これらの新たに生成された増幅乱数ビット列は、非線形変換部８０に出力され、非線形変換が行われる（ステップＳ１８）。非線形変換部８０は、予め設定された非線形関数に基づいてＮｏビットのαβ個の増幅乱数ビット列を非線形変換し、Ｎｏビットを有する１個の乱数ビット列を疑似乱数として出力する。

非線形変換部８０から疑似乱数を出力すると、再びステップＳ１４まで戻り、ステップＳ１４からステップＳ１８までの処理を繰り返し行う。そして、平文から暗号文に変換するために必要な分の疑似乱数を発生させる。

上述の疑似乱数発生器１によると、選択用乱数ビット列出力手段５１によって出力したＮｉビットの選択用乱数ビット列に基づき乱数テーブルを参照することで、Ｎｉビットよりも大きなビット数を有するＮｏビットの増幅乱数ビット列を非線形変換部８０に供給することができる。したがって、従来はボトルネックとなっていた非線形変換部８０よりも上流側のスループットを向上させることができ、非線形変換部８０のスループットに近づけることができる。したがって、疑似乱数発生器１全体のスループットを高速化することができる。

また、選択用乱数ビット列出力部２０からの選択用乱数ビット列の入力に応じて、乱数テーブル順番入れ替え処理を行うので、疑似乱数の暗号強度を増大させることができる。特に、本実施の形態では、乱数テーブル６２_{1}～$６２_{\alpha\beta}$の組合せパターンを（αβ）の階乗個（以下、階乗を「！」で表す）にすることができ

る。したがって、乱数テーブル部６１の内容を既知と仮定したときに成立する攻撃では、（２＾（αβ×Ｎｉ））×（αβ）！の計算量が必要となり、この計算量は、Ｌｋビットの秘密鍵を全数探索する場合の計算量よりも多くなることから、十分な暗号強度を備えていることがわかる。

また、上述の疑似乱数発生器１は、一の選択用乱数ビット列出力手段５１から出力した乱数ビット列を用いて複数（β個）の乱数テーブルを参照し、各乱数テーブルから選択した乱数ビット列に排他的論理和演算を施す処理を行っている。したがって、乱数テーブル６１から読み出した増幅乱数ビット列をそのまま非線形変換部８０に出力した場合のように暗号強度が増幅乱数ビット列発生手段６６そのものに依存するのを防ぎ、暗号強度を更に向上させている。

次に、本実施の形態における具体的な一実施例について説明する。第８図は、本実施例の疑似乱数発生器１を概略的に示す概念図、第９図は、乱数テーブル部６１を概略的に示す概念図である。尚、本実施例では、各設定値（パラメータ）を以下のように設定した場合を例に説明する。

選択用乱数ビット列出力手段の数：８個（α＝８）

各選択用乱数ビット列出力手段に対応した乱数テーブルの数：２個（β＝２）

乱数テーブルのインデックス部の長さ：２＾８個（Ｎｉ＝８）

乱数テーブルの乱数ビット列部の長さ：２＾１６個（Ｎｏ＝１６）

秘密鍵の長さ：１２８ビット（Ｌｋ＝１２８）

非線形変換部８０の非線形関数ｆ（ｘ）：

f(x) = x1 + x5
+ x1x2 + x1x3 + x2x3 + x2x5 + x2x6 + x3x6
+ x1x7 + x2x7 + x4x8 + x5x8
+ x1x2x3 + x1x2x4 + x1x3x4 + x2x3x4 + x1x2x5
+ x2x4x5 + x3x4x5 + x1x2x6 + x2x3x6 + x1x4x6
+ x4x5x6 + x1x2x7 + x2x3x7 + x1x4x7 + x1x5x7
+ x2x5x7 + x4x5x7 + x1x6x7 + x4x6x7 + x5x6x7
+ x1x2x8 + x1x3x8 + x2x3x8 + x3x4x8 + x1x5x8
+ x3x5x8 + x4x5x8 + x3x6x8 + x4x6x8 + x5x6x8

+ x1x7x8 + x2x7x8
+ x1x2x4x5 + x1x3x4x5 + x2x3x4x5 + x1x2x4x6
+ x1x3x4x6 + x2x3x4x6 + x1x4x5x6 + x2x4x5x6
+ x3x4x5x6 + x1x2x3x7 + x1x2x4x7 + x2x3x4x7
+ x1x2x5x7 + x1x4x5x7 + x2x4x5x7 + x1x2x6x7
+ x1x3x6x7 + x2x3x6x7 + x1x4x6x7 + x2x4x6x7
+ x3x4x6x7 + x1x5x6x7 + x2x5x6x7 + x3x5x6x7
+ x1x2x4x8 + x1x2x5x8 + x1x3x5x8 + x1x4x5x8
+ x1x2x6x8 + x2x3x6x8 + x1x4x6x8 + x2x5x6x8
+ x3x5x6x8 + x1x3x7x8 + x1x4x7x8 + x2x4x7x8
+ x3x4x7x8 + x2x5x7x8
+ x1x2x3x4x5 + x1x2x3x4x6 + x1x3x4x5x6
+ x2x3x4x5x6 + x1x2x4x5x7 + x2x3x4x5x7
+ x1x2x4x6x7 + x1x3x4x6x7 + x1x4x5x6x7
+ x2x4x5x6x7 + x1x2x3x4x8 + x1x2x3x5x8
+ x1x2x4x5x8 + x1x2x3x6x8 + x1x2x4x6x8
+ x1x3x4x6x8 + x2x3x5x6x8 + x1x4x5x6x8
+ x2x4x5x6x8 + x1x2x3x7x8 + x1x3x4x7x8
+ x1x3x5x7x8 + x2x3x5x7x8 + x3x4x5x7x8
+ x1x3x6x7x8 + x3x4x6x7x8
+ x1x2x3x4x5x8 + x1x2x3x4x6x8
+ x1x3x4x5x6x8 + x2x3x4x5x6x8
+ x1x2x3x4x7x8 + x1x2x3x5x7x8
+ x1x2x4x5x7x8 + x1x3x4x5x7x8
+ x1x3x4x6x7x8 + x2x3x4x6x7x8
+ x1x2x5x6x7x8 + x1x3x5x6x7x8

本実施例では、各選択用乱数ビット列出力手段５１が、ユーザから与えられる秘密鍵Ｋに基づいて線形フィードバックシフトレジスタ５３の再構成を行い、その再構成後の線形フィードバックシフトレジスタ５３′を用いて選択用乱数ビッ

ト列を出力するように構成されている。

まず最初に、この選択用乱数ビット列出力手段５１の構成及びその動作について説明する。選択用乱数ビット列出力手段５１は、第８図に示すように、初期値設定手段１２、線形フィードバックシフトレジスタ５３、線形フィードバックシフトレジスタ再構成手段１４を有している。

初期値設定手段１２は、ユーザから与えられる秘密鍵Kに基づいて初期値を設定するものであり、秘密鍵Kをビット列に変換し、初期値として線形フィードバックシフトレジスタ５３のシフトレジスタ内に割り当てるように構成されている。初期値設定手段１２は、本実施例では、ＲＣ４　Ｓｙｐｐｅｔｒｉｃ　Ｓｔｒｅａｐ　Ｃｉｐｈｅｒ（ＲＳＡ　Ｄａｔａ　Ｓｅｃｕｒｉｔｙ　Ｉｎｃ.製）を用いており、増幅乱数ビット列発生手段６６と共用している。

線形フィードバックシフトレジスタ５３は、従来技術で説明したものと同様に、１ビットの情報を記憶できるｎ個のシフトレジスタと、排他的論理和演算回路を有している。そして、本実施の形態では、１周期分のビット数ｍが（２＾ｎ）－１個となるビット列、いわゆるM系列を出力可能な構成に予め設定されている。

第１１図は、本実施の形態における線形フィードバックシフトレジスタ５３の初期多項式を例示するものである。初期多項式は、M系列を出力するように予め設定されている特性多項式であり、１項目の指数部分がシフトレジスタの個数を示し、２項目以降の指数部分が排他的論理和演算回路に接続された結線位置を示している。例えば、１段目の線形フィードバックシフトレジスタ（ＬＦＳＲ１）５３は、１２９個のシフトレジスタを有し、８０番目、８番目、１番目のシフトレジスタがフィードバックタップによって排他的論理和演算回路に接続されていることを示している。尚、本実施の形態では、シフトレジスタの個数ｎは、全て素数個に設定されている。

線形フィードバックシフトレジスタ再構成手段１４は、線形フィードバックシフトレジスタ５３の構成を秘密鍵Kによって動的に変更して再構成するものである。具体的には、出力系列がM系列のビット列をｓ個ごとにサンプルした新ビット列は、M系列の１周期分のビット数ｍ（＝（２＾ｎ）－１）と導出値ｓとが互いに素であるとき、すなわち、１以外の共通の約数を持たないときは、他の構成

を有する線形フィードバックシフトレジスタのM系列になり、また、バーレイキャンプマッセイアルゴリズムによって、少なくとも２周期分以上のビット数を有するビット列から、そのビット列を出力可能な等価で最小の線形フィードバックシフトレジスタの特性多項式を求めることができることを利用して、線形フィードバックシフトレジスタ５３の再構成を行う。

線形フィードバックシフトレジスタ再構成手段１４は、初期値設定手段１２によって与えられた初期値から導出値ｓを算出し、導出値ｓと線形フィードバックシフトレジスタ５３の１周期分のビット数ｍ（＝（２＾ｎ）－１）を２倍した値２ｍとを乗算し、線形フィードバックシフトレジスタ５３から出力させるビット列のビット数２ｍｓを算出する。

そして、初期値をもとに線形フィードバックシフトレジスタ５３から２ｍｓ個のビット列を出力させ、その２ｍｓ個のビット列から導出値ｓの間隔ごとにビット列を取り出して新ビット列を生成し、その新ビット列を用いてバーレイキャンプマッセイアルゴリズムにより線形フィードバックシフトレジスタ５３の構成を変更する。

尚、線形フィードバックシフトレジスタ５３から出力させるビット列のビット数は、新ビット列のビット数が２ｍ個以上であれば、等価な最小の線形フィードバックシフトレジスタを求めることができるので、２ｍｓ個以上であればよい。

バーレイキャンプマッセイアルゴリズムとは、線形フィードバックシフトレジスタ５３のシフトレジスタの個数ｎ（線形複雑度）の２倍以上のビット数を有するビット列を入手することで、そのビット列を出力可能な等価な最小の線形フィードバックシフトレジスタを得ることができるというアルゴリズムである。バーレイキャンプマッセイアルゴリズムについては、例えば、文献１「暗号理論入門（第２版）」、共立出版社、岡本栄司著、２００２年４月１０日発行、に詳細に説明されている。

第１２図は、線形フィードバックシフトレジスタ５３の再構成処理を説明するフローチャートである。まず最初に、初期値設定手段１２によって初期値が設定される（ステップＳ４１）。初期値は、利用者から与えられるＬｋビットの秘密鍵Ｋに基づいて設定される。初期値設定手段１２において秘密鍵Ｋから初期値が

設定されると、その初期値は、線形フィードバックシフトレジスタ５３のシフトレジスタ内にセットされる。

次に、所定の演算処理により初期値から線形フィードバックシフトレジスタ５３の１周期分のビット数mと互いに素である導出値ｓを算出する（ステップＳ４２）。導出値ｓは、初期値に対して、例えばMD５（Message Digest 5）などのハッシュ関数を施してハッシュ値を求め、そのハッシュ値に最も近似した素数が採用される。導出値ｓは、初期値から求めることができ、かつビット数mと互いに素であればよく、上記の算出方法によって求められるものに限定されない。但し、秘匿性を維持するために、上記所定の演算処理は、一方向性を満足しうる演算処理でなければならない。

導出値ｓを算出すると、次に、線形フィードバックシフトレジスタ５３から出力させるビット列のビット数２ｍｓを算出する（ステップＳ４３）。線形フィードバックシフトレジスタ５３から出力させるビット列のビット数２ｍｓは、線形フィードバックシフトレジスタ５３の１周期分のビット数m（＝（２＾ｎ）－１）を２倍した値と、導出値ｓとを乗算することによって求められる。

そして次に、線形フィードバックシフトレジスタ５３から初期値をもとに２ｍｓ個のビット数を有するビット列を出力させ（ステップＳ４４）、そのビット列から新ビット列を生成する（ステップＳ４５）。新ビット列は、２ｍｓ個のビット列から導出値ｓの間隔ごとに取り出したビット列によって構成され、そのビット数は２m個となる。

ここで、出力系列がM系列のビット列をｓ個ごとにサンプルしたビット列は、そのM系列の１周期分のビット数mと導出値ｓとが互いに素であれば、他の構成を有する線形フィードバックシフトレジスタのM系列となることから、この新ビット列も、M系列となる。

そして、その新ビット列に基づいて線形フィードバックシフトレジスタ５３の構成を再構成する（ステップＳ４６）。線形フィードバックシフトレジスタ５３の再構成は、バーレイキャンプマッセイアルゴリズムを用いて行われる。バーレイキャンプマッセイアルゴリズムによれば、少なくとも２周期分以上のビット数を有するビット列があれば、かかるビット列を出力可能な等価で最小の線形フィ

ードバックシフトレジスタ５３を求めることができるので、２m個のビット数を有する新ビット列から新たな線形フィードバックシフトレジスタ５３の特性多項式を導出して、再構成を行う。

再構成後の線形フィードバックシフトレジスタ５３′は、再構成前と同一の次数及び異なる結線の特性多項式を有し、同一の初期値を与えた場合に、再構成前と異なるM系列を出力可能な構成を有する。

線形フィードバックシフトレジスタ再構成手段１４による線形フィードバックシフトレジスタ５３の再構成が終了すると、再構成された線形フィードバックシフトレジスタ５３′から初期値をもとに乱数ビット列を発生させる処理が行われる（ステップＳ４７）。これにより、乱数ビット列出力部５０からは再構成前とは異なるM系列の乱数ビット列が出力される。

尚、上記のステップＳ４６で、新ビット列に基づいて線形フィードバックシフトレジスタ５３の構成を再構成する代わりに、新ビット列を出力可能な構成を有する第２の線形フィードバックシフトレジスタを生成し、ステップＳ４７で、その第２の線形フィードバックシフトレジスタによって初期値をもとに乱数ビット列を発生させてもよい。これによれば、線形フィードバックシフトレジスタ５３を２つに分けることができ、より秘匿性の向上を図ることができる。

上記の選択用乱数ビット列出力手段５１は、線形フィードバックシフトレジスタ５３の構成を初期値に基づいて容易かつ動的に変更することができ、変更後もM系列を出力させることができる。したがって、攻撃者は、再構成前の線形フィードバックシフトレジスタ５３の構成を取得することができない。これにより、従来、線形フィードバックシフトレジスタ５３の構成が既知であることを前提に成り立っていた既存の暗号解読法は、成立しなくなる。したがって、高い暗号強度を得ることができ、情報の秘匿性を保つことができる。

次に、上記の選択用乱数ビット列出力手段５１を備えた疑似乱数発生器１による疑似乱数発生方法について説明する。第１０図は、本実施例における疑似乱数発生方法を説明するフローチャートである。

まず最初に、乱数ビット列出力部５０は、ユーザから１２８ビット（Ｌｋ＝１２８）を有する任意の秘密鍵Ｋの入力を受け取ると、その秘密鍵Ｋに基づいて再

構成前の線形フィードバックシフトレジスタ５３の初期値を設定する（ステップＳ２１）。

そして、その初期値に基づいて線形フィードバックシフトレジスタ５３を再構成し（ステップＳ２２）、再構成後の線形フィードバックシフトレジスタ５３′に初期値を設定する（ステップＳ２３）。この初期値の設定を、全ての乱数ビット列出力手段11_1～11_8について行う。

次に、乱数ビット列増幅部６０は、乱数テーブル部６１の初期設定を行う（ステップＳ２４）。ここでは、まず、増幅乱数ビット列発生手段６６に秘密鍵Ｋを与え、高速で乱数ビット列を発生させる処理が行われるが、本実施例では、上述のように、増幅乱数ビット列発生手段６６と選択用乱数ビット列出力手段５１の初期値設定手段１２とを共用しているので、別途出力することはせずに、線形フィードバックシフトレジスタ５３の初期値として出力した乱数ビット列をそのまま用いる。

乱数テーブル初期設定手段６５は、その乱数ビット列を１６ビット（Ｎｏ＝１６）ごとに分割し、増幅乱数ビット列として各乱数テーブル62_1～62_{16}の全ての乱数ビット列格納部Ｒｏに順次格納する。

以上の初期値設定段階が終了すると（ステップＳ２１～Ｓ２４）、待機状態となる。そして、平文の暗号化装置（従来技術を参照）への入力をトリガとして、疑似乱数を発生させる処理（ステップＳ２５～Ｓ２７）に移行する。

ここでは、各選択用乱数ビット列出力手段51_1～51_8ごとにそれぞれ選択用乱数ビット列を出力させ、乱数ビット列増幅部６０のバッファ内にそれぞれ記憶させる処理が行われる。具体的には、各選択用乱数ビット列出力手段51_1～51_8から８ビットの選択用乱数ビット列がそれぞれ出力され（ステップＳ２７）、その数が各選択用乱数ビット列出力手段１に対して２個分（$\beta=2$）であり（ステップＳ２６でＹｅｓ）、各選択用乱数ビット列出力手段51_1～51_8にそれぞれ対応する分である場合（ステップＳ２５でＹｅｓ）には、必要数の選択乱数ビット列が得られたとして次の乱数ビット列増幅段階に移行する。したがって、ここまでの処理により、バッファ内には８ビットを有する１６個の選択用乱数ビット列が記憶される。

次に、秘密鍵Ｋ０に基づき入れ替え用乱数発生手段６８により１６個の入れ替え用乱数を発生させ（ステップＳ２８）、乱数テーブルの順番入れ替え処理が行われる（ステップＳ２９）。ここでは、１６個の乱数が順番入れ替え用のテーブル番号として、乱数テーブル62_1～62_{16}に付与される。したがって、１番～１６番までのテーブル番号が順不同で乱数テーブル62_1～62_{16}に付与される。そして、その付与されたテーブル番号をもとに各乱数テーブル62_1～62_{16}の順番の入れ替えを行う。ここでは、選択用乱数ビット列出力手段51_1～51_nに対してテーブル番号が１番～１６番に順番に並ぶように降順に入れ替える処理が行われる。これにより、乱数テーブル部６１内の増幅乱数ビット列は、その順番が乱数テーブル単位でランダムに入れ替えられる。

そして次に、各乱数テーブル62_1～62_{16}内から該当する増幅乱数ビット列を選択する処理が行われる（ステップＳ３０～Ｓ３２）。例えば、選択用乱数ビット列11_1から出力されバッファに記憶された１番目の選択用乱数ビット列を引数として乱数テーブル62_1が参照される（ステップＳ３２）。そして、その引数と等しい値を有するインデックス番号を選択し、そのインデックス番号に対応する乱数ビット列格納部Ｒｏに格納された増幅乱数ビット列を選択する。

例えば、選択用乱数ビット列出力手段51_1から出力され乱数テーブル62_1に対応するものとしてバッファに記憶された選択用乱数ビット列が「０００００１１」である場合、これを８桁の２進数とみなし、１０進数に変換して引数「３」を得る。この引数「３」を用いて乱数テーブル62_1を参照し、インデックス部Ｒｏのインデックス番号が「３」の乱数ビット列格納部Ｒｏに格納されている増幅乱数ビット列「０１０１１０１０１１０１１１０１１０」を選択する。

そして、乱数テーブル62_1と乱数テーブル62_2からそれぞれ増幅乱数ビット列を選択すると（ステップＳ３１でＹｅｓ）、これら２つの増幅乱数ビット列の排他的論理和演算処理を行い（ステップＳ３３）、１６ビットを有する１個の新たな増幅乱数ビット列を生成する。

そして、同様の処理を各乱数テーブル62_3～62_{16}について行い（ステップＳ３０でＹｅｓ）、合計で８個の新たな増幅乱数ビット列を生成すると、非線形変換部８０に出力して、非線形変換段階に移行する。

非線形変換部８０では、乱数ビット列増幅部６０よりこれらのＮｏビットを有する８個の増幅乱数ビット列を入力すると、非線形関数ｆ（ｘ）により非線形変換し（ステップＳ３４）、１６ビットを有する１個の乱数ビット列を得る。そして、上記ステップＳ２５～ステップＳ３４の処理を繰り返し実行して必要数の疑似乱数を得る。

本実施例については、処理速度の高速化及び乱数性が適切に確保されているかについて実験を行っており、その結果、従来と比較して１７０倍も処理速度を向上でき、同時に適切な乱数性も確保されているとの結果を得た。以下に、その実験内容及び実験結果について説明する。

実験に使用したコンピュータは、ＣＰＵ：Ｐｅｎｔｉｕｍ（登録商標）４（１．７ＧＨｚ）、メモリ：２５６ＭＢである。また、各設定値は、上述の実施例と同一のものとする。そして、入れ替え用乱数ビット列発生手段２８に用いられる秘密鍵Ｋ０は、１６進数表記で以下のものに固定した状態として実験を行った。

$K0 = (f1e2d3c4b5a69788796a5b4c3d2e1f10)_{16}$

第１３図は、スループットの計測結果を示す表である。表中の従来型とは、８個の線形フィードバックシフトレジスタ５３と、非線形変換部８０を用いて構成した、第１７図に示すような従来の非線形コンバイナ型疑似乱数発生器を示す。

本実験結果によれば、第１３図に示すように、疑似乱数発生器１の平均スループットが、線形フィードバックシフトレジスタ５３そのものの平均スループットから、非線形変換部８０の平均スループットに向上しており、更に、従来型の約１７０倍(116.4Mbps/sec÷0.680Mbps/sec=171.176...)になっていることがわかる。したがって、このスループット計測結果から、乱数テーブル６２を用いたことが疑似乱数発生器１の高速化に有効であることがわかる。

本実施例における疑似乱数発生器１のスループットは、次式（１）で表される。

【数式】

$$\frac{1}{T}=\frac{N_I}{N_O}\left(\frac{n}{T_1}+\frac{1}{T_2}+\frac{1}{T_3}\right)+\frac{nm}{T_4}+\frac{1}{T_5}. \quad (1)$$

Ｔ１は、１つの線形フィードバックシフトレジスタ５３の平均スループットを示し、Ｔ２は、ＲＣ４（増幅乱数ビット列発生手段６６）の平均スループットを

示す。また、Ｔ３は、乱数テーブル入れ替え手段６７による乱数テーブル入れ替え処理の平均スループットを示し、Ｔ４は、１つの乱数テーブル６２の平均スループットを示す。そして、Ｔ５は、非線形変換部８０の平均スループットを示す。上記式（１）から乱数テーブル６２の計算量が無視できると仮定すると、Ｎｏビット／Ｎｉビットを小さくするほど非線形変換部８０のスループットに近づけることができ、高速化を図ることができる。

疑似乱数の暗号強度の検証については、ＮＩＳＴという疑似乱数検定ツールを用いて検定を行った。ＮＩＳＴとは、物理乱数及び疑似乱数生成器からの出力データについて乱数性のテストを行うツールであり、１６項目からのテストからなる統計のパッケージである。ＮＩＳＴについては、http://crsc.nist.gov/rug.に詳しく説明されている。第１４図は、本検定に使用したＮＩＳＴのパラメータを示す表である。各種テストを行うことによって出力されたp-valueが０＜p-value＜１を満たす場合に、そのテスト項目をパスしたとみなしている。本実施例による疑似乱数発生器１の疑似乱数を検定したところ、全てのテスト項目をパスしていることが確認できた。第１５図は、今回の実験によるＮＩＳＴの検定結果を示す図である。

尚、上述の実施例に示した各設定値は、暗号の安全性を確認するために設定したものであり、これに限定されるものではない。また、本発明は、上述の実施の形態に限定されるものではなく、本発明の趣旨を逸脱しない範囲で種々の変更、組み合わせが可能である。

［発明の効果］

以上説明したように、本発明に係る疑似乱数発生方法によれば、出力系列がM系列のビット列をｓ個ごとにサンプルしたビット列は、そのM系列の１周期分のビット数ｍ（＝（２＾ｎ）－１）と導出値ｓが互いに素であるときには、他の構成を有する線形フィードバックシフトレジスタのM系列を構成し、また、バーレイキャンプマッセイアルゴリズムによって、少なくとも２周期分以上のビット数を有するビット列から線形フィードバックシフトレジスタを求めることができることを利用して、線形フィードバックシフトレジスタの構成を初期値に基づいて動的に変更することができ、変更後の線形フィードバックシフトレジスタからM

系列のビット列を出力させることができる。

したがって、解読者は、疑似乱数発生器から出力される疑似乱数に基づいて再構成前の線形フィードバックシフトレジスタの構成を得ることができず、初期値や秘密鍵も解読することができない。この結果、高い暗号強度を得ることができ、情報の秘匿性を保つことができる。

また、他の発明によれば、秘密鍵に基づいて所定のビット数を有する選択用乱数ビット列を出力させ、その選択用乱数ビット列を用いて乱数テーブルを参照することにより、乱数テーブル内の複数の増幅乱数ビット列の中から該当する増幅乱数ビット列を選択し、非線形変換手段によって非線形変換して疑似乱数として出力させるので、小さなビット列を有する選択用乱数ビット列に基づいて、より大きなビット数を有する増幅乱数ビット列を得ることができる。

したがって、非線形変換手段に入力される乱数ビット列をより大きなビット数を有するものにすることができる。これにより、従来、ボトルネックとなっていた非線形変換手段よりも上流側の乱数ビット列を出力する部分のスループットを向上させ、非線形変換手段のスループットに近づけることができ、疑似乱数発生器全体のスループットを高速化することができる。

[図面の簡単な説明]

[第１図]

本実施の形態における疑似乱数発生器を説明する図である。

[第２図]

本実施の形態における線形フィードバックシフトレジスタの初期多項式を例示するものである。

[第３図]

本実施の形態における疑似乱数発生器の動作を説明するフローチャートである。

[第４図]

疑似乱数発生器の機能を概略的に示す説明図である。

[第５図]

乱数テーブル部の説明図である。

[第６図]

乱数ビット列増幅部内に構成される各構成要素を説明する概念図である。

［第７図］

本実施の形態における疑似乱数発生方法を説明するフローチャートである。

［第８図］

本実施例における疑似乱数発生器を概略的に示す概念図である。

［第９図］

乱数テーブル部を概略的に示す概念図である。

［第１０図］

本実施例における疑似乱数発生方法を説明するフローチャートである。

［第１１図］

本実施の形態における線形フィードバックシフトレジスタの初期多項式を例示するものである。

［第１２図］

線形フィードバックシフトレジスタの再構成処理を説明するフローチャートである。

［第１３図］

スループットの計測結果を示す表である。

［第１４図］

本検定に使用したＮＩＳＴのパラメータを示す表である。

［第１５図］

ＮＩＳＴの検定結果を示す図である。

［第１６図］

従来の逐次暗号方式を説明する図である。

［第１７図］

暗号化装置の疑似乱数発生器を説明する図である。

［第１８図］

一般的な線形フィードバックシフトレジスタの構成を簡単に説明する図である。

［符号の説明］

1　疑似乱数発生器

１０　疑似乱数生成部
１１　線形フィードバックシフトレジスタ
１２　線形フィードバックシフトレジスタ再構成手段
２０　非線形変換部
５０　乱数ビット列出力部
５１　選択用乱数ビット列出力手段
５２　初期値設定手段
５３　線形フィードバックシフトレジスタ
５４　線形フィードバックシフトレジスタ再構成手段
６０　乱数ビット列増幅部
６１　乱数テーブル部
62_1～$62_{\alpha\beta}$　乱数テーブル
６３　排他的論理和演算処理手段
６４　増幅乱数ビット列選択手段
６５　乱数テーブル初期設定手段
６６　増幅乱数ビット列発生手段
６７　乱数テーブル順番入れ替え手段
６８　入れ替え用乱数発生手段
７０　非線形変換部

請求の範囲

1． n個のシフトレジスタを有し、1周期分のビット数が（2＾n）－1個となるビット列を出力可能な線形フィードバックシフトレジスタの初期値を設定する第1ステップと、

所定の演算処理により前記初期値から前記線形フィードバックシフトレジスタの1周期分のビット数と互いに素である導出値を求める第2ステップと、

該導出値と前記線形フィードバックシフトレジスタの1周期分のビット数を2倍以上した値とを乗算して、前記第1線形フィードバックシフトレジスタにより出力させるビット列のビット数を算出する第3ステップと、

前記算出したビット数分のビット列を前記線形フィードバックシフトレジスタから前記初期値をもとに出力させる第4ステップと、

該出力したビット列から前記導出値の間隔ごとにビット列を取り出して新ビット列を生成する第5ステップと、

該新ビット列を出力可能な構成に前記線形フィードバックシフトレジスタの構成を再構成する第6ステップと、

該再構成した後の線形フィードバックシフトレジスタから前記初期値をもとに疑似乱数を発生させる第7ステップと、

を有することを特徴とする疑似乱数発生方法。

2． 前記初期値に対してハッシュ関数を施してハッシュ値を求め、該ハッシュ値に最も近似した素数を導出値として採用することを特徴とする請求項1に記載の疑似乱数発生方法。

3． 前記線形フィードバックシフトレジスタの再構成は、バーレイキャンプマッセイアルゴリズムを用いて行われることを特徴とする請求項1または2に記載の疑似乱数発生方法。

4． 前記第7ステップで発生させた疑似乱数を非線形変換する第8ステップ

を有することを特徴とする請求項１～３のいずれかに記載の疑似乱数発生方法。

５．　ｎ個のシフトレジスタを有し、１周期分のビット数が（２＾ｎ）－１個となるビット列を出力可能な線形フィードバックシフトレジスタと、

秘密鍵に基づき前記線形フィードバックシフトレジスタの初期値を設定する初期値設定手段と、

所定の演算処理により前記初期値から前記線形フィードバックシフトレジスタの１周期分のビット数と互いに素である導出値を求める導出値算出手段と、

前記線形フィードバックシフトレジスタの１周期分のビット数を２倍以上した値と前記導出値とを乗算して、前記第１線形フィードバックシフトレジスタにより出力させるビット列のビット数を算出するビット数算出手段と、

前記算出したビット数分のビット列を前記線形フィードバックシフトレジスタから前記初期値をもとに出力させるビット列出力手段と、

該出力したビット列から前記導出値の間隔ごとにビット列を取り出して新ビット列を生成する新ビット列生成手段と、

該新ビット列を出力可能な構成に前記線形フィードバックシフトレジスタの構成を再構成する線形フィードバックシフトレジスタ再構成手段と、

該再構成後の線形フィードバックシフトレジスタから前記初期値をもとに疑似乱数を発生させる疑似乱数発生手段と、を有することを特徴とする疑似乱数発生器。

６．　前記線形フィードバックシフトレジスタ再構成手段の代わりに、新ビット列を出力可能な構成を有する第２の線形フィードバックシフトレジスタを生成する線形フィードバックシフトレジスタ生成手段を設け、

前記疑似乱数発生手段は、前記第２の線形フィードバックシフトレジスタによって初期値をもとに疑似乱数を発生させることを特徴とする請求項５に記載の疑似乱数発生器。

７．　秘密鍵に基づいて所定ビット数を有する選択用乱数ビット列を出力する

選択用乱数ビット列出力手段と、

前記選択用乱数ビット列よりも大きなビット数を有する増幅乱数ビット列を予め複数格納した乱数テーブルと、

前記選択用乱数ビット列出力手段から出力された選択用乱数ビット列を用いて前記乱数テーブルを参照することにより、前記乱数テーブル内の複数の増幅乱数ビット列の中から該当する増幅乱数ビット列を選択可能な増幅乱数ビット列選択手段と、

該増幅乱数ビット列選択手段により選択された増幅乱数ビット列を非線形関数によって非線形変換し疑似乱数として出力する非線形変換手段と、

を有することを特徴とする疑似乱数発生器。

８． 前記秘密鍵が与えられることにより前記秘密鍵に基づいて前記増幅乱数ビット列を発生させ、前記乱数テーブルに格納して、前記乱数テーブルの初期設定を行う乱数テーブル初期設定手段を有することを特徴とする請求項７に記載の疑似乱数発生器。

９． 前記選択用乱数ビット列出力手段は、複数設けられ、

前記乱数テーブルは、前記各選択用乱数ビット列出力手段にそれぞれ対応するように設けられ、

前記増幅乱数ビット列選択手段は、前記各選択用乱数ビット列出力手段から各々出力された前記各選択用乱数ビット列を用いて前記各選択用乱数ビット列出力手段ごとに対応する前記乱数テーブルをそれぞれ参照し、前記各乱数テーブル内からそれぞれ該当する増幅乱数ビット列を選択し、

非線形変換手段は、前記各増幅乱数ビット列選択手段によって前記各乱数テーブルから選択された前記各増幅乱数ビット列を用いて非線形関数により非線形変換し疑似乱数として出力することを特徴とする請求項７または８に記載の疑似乱数発生器。

１０． 前記各選択用乱数ビット列出力手段ごとにそれぞれ複数の乱数テーブ

ルを設け、

前記増幅乱数ビット列選択手段により前記各乱数テーブル内から選択された各増幅乱数ビット列を前記選択用乱数ビット列出力手段ごとに排他的論理和演算して非線形変換手段に出力する排他的論理和演算処理手段を有することを特徴とする請求項９に記載の疑似乱数発生器。

１１．　所定のタイミングで前記各乱数テーブル同士の入れ替えを行う乱数テーブル入れ替え手段を有することを特徴とする請求項９または１０に記載の疑似乱数発生器。

１２．　前記乱数テーブル入れ替え手段は、

前記各乱数テーブルを参照するために必要な選択用乱数ビット列を前記選択用乱数ビット列出力手段が出力するごとに、前記各乱数テーブル同士の入れ替えを行うことを特徴とする請求項１１に記載の疑似乱数発生器。

１３．　前記乱数テーブル入れ替え手段は、

前記各乱数テーブルの個数と等しい個数の乱数テーブル入れ替え用乱数を発生させ、

該乱数テーブル入れ替え用乱数を前記乱数テーブルのテーブル番号として各乱数テーブルに付与し、前記テーブル番号をもとに予め設定された規則に従って前記乱数テーブルの順番を入れ替えることを特徴とする請求項１１または１２に記載の疑似乱数発生器。

１４．　コンピュータを、

秘密鍵に基づいて所定ビット数を有する選択用乱数ビット列を出力させる選択用乱数ビット列出力手段と、

前記選択用乱数ビット列よりも大きなビット数を有する増幅乱数ビット列を予め複数格納した乱数テーブルと、

前記選択用乱数ビット列出力手段から出力された選択用乱数ビット列を用いて

前記乱数テーブルを参照することにより、前記乱数テーブル内の複数の増幅乱数ビット列の中から該当する増幅乱数ビット列を選択可能な増幅乱数ビット列選択手段と、

該増幅乱数ビット列選択手段により選択された増幅乱数ビット列を非線形関数によって非線形変換し疑似乱数として出力する非線形変換手段として機能させるための疑似乱数発生プログラム。

１５．　前記秘密鍵が与えられることにより前記秘密鍵に基づいて前記増幅乱数ビット列を発生させ、前記乱数テーブルに格納して、前記乱数テーブルの初期設定を行う乱数テーブル初期設定手段としてコンピュータを機能させることを特徴とする請求項１４に記載の疑似乱数発生プログラム。

１６．　前記選択用乱数ビット列出力手段は、複数設けられ、

前記乱数テーブルは、前記各選択用乱数ビット列出力手段にそれぞれ対応するように設けられ、

前記増幅乱数ビット列選択手段は、前記各選択用乱数ビット列出力手段から各々出力された前記各選択用乱数ビット列を用いて前記各選択用乱数ビット列出力手段ごとに対応する前記乱数テーブルをそれぞれ参照し、前記各乱数テーブル内からそれぞれ該当する増幅乱数ビット列を選択し、

非線形変換手段は、前記各増幅乱数ビット列選択手段によって前記各乱数テーブルから選択された前記各増幅乱数ビット列を用いて非線形関数により非線形変換し疑似乱数として出力することを特徴とする請求項１４または１５に記載の疑似乱数発生プログラム。

１７．　前記各選択用乱数ビット列出力手段ごとにそれぞれ複数の乱数テーブルを設け、

前記増幅乱数ビット列選択手段により前記各乱数テーブル内から選択された各増幅乱数ビット列を前記選択用乱数ビット列出力手段ごとに排他的論理和演算して非線形変換手段に出力する排他的論理和演算処理手段としてコンピュータを機

能させることを特徴とする請求項１６に記載の疑似乱数発生プログラム。

１８．　所定のタイミングで前記各乱数テーブル同士の入れ替えを行う乱数テーブル入れ替え手段としてコンピュータを機能させることを特徴とする請求項１６または１７に記載の疑似乱数発生プログラム。

１９．　前記乱数テーブル入れ替え手段は、

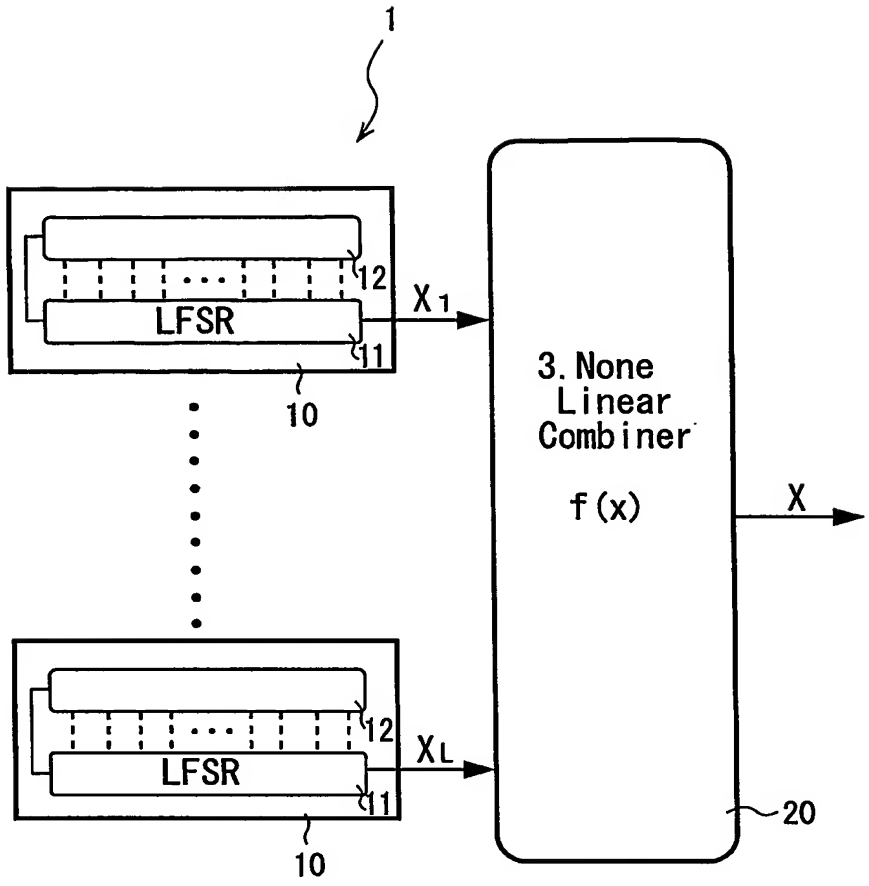
前記各乱数テーブルを参照するために必要な選択用乱数ビット列を前記選択用乱数ビット列出力手段が出力するごとに、前記各乱数テーブル同士の入れ替えを行うことを特徴とする請求項１８に記載の疑似乱数発生プログラム。

２０．　前記乱数テーブル入れ替え手段は、

前記各乱数テーブルの個数と等しい個数の乱数テーブル入れ替え用乱数を発生させ、

該乱数テーブル入れ替え用乱数を前記乱数テーブルのテーブル番号として各乱数テーブルに付与し、前記テーブル番号をもとに予め設定された規則に従って前記乱数テーブルの順番を入れ替えることを特徴とする請求項１８または１９に記載の疑似乱数発生プログラム。

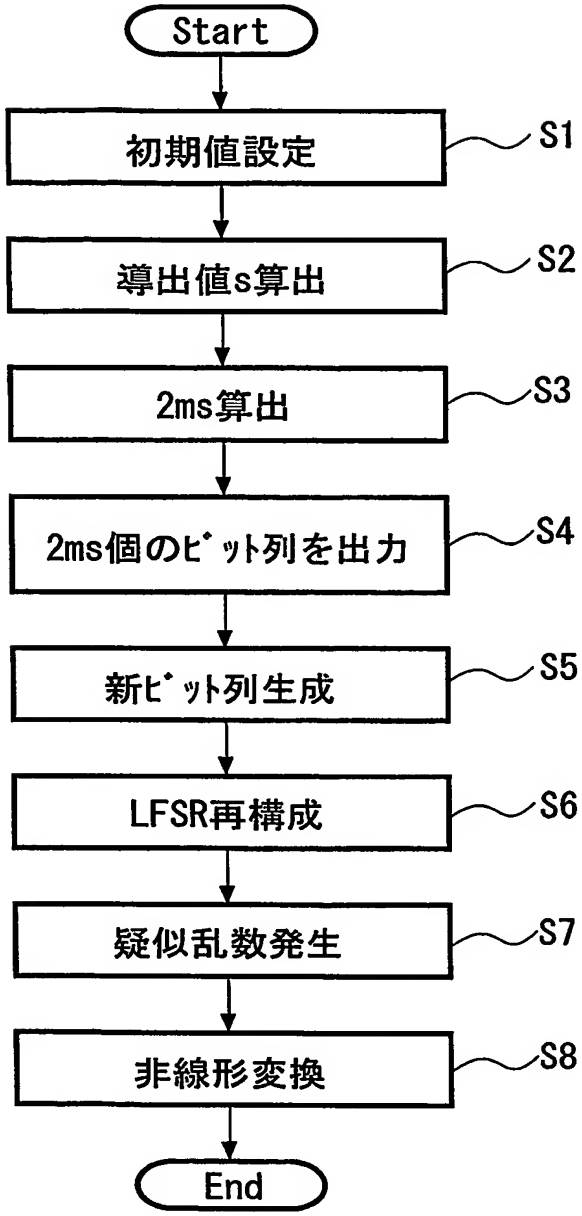
第1図



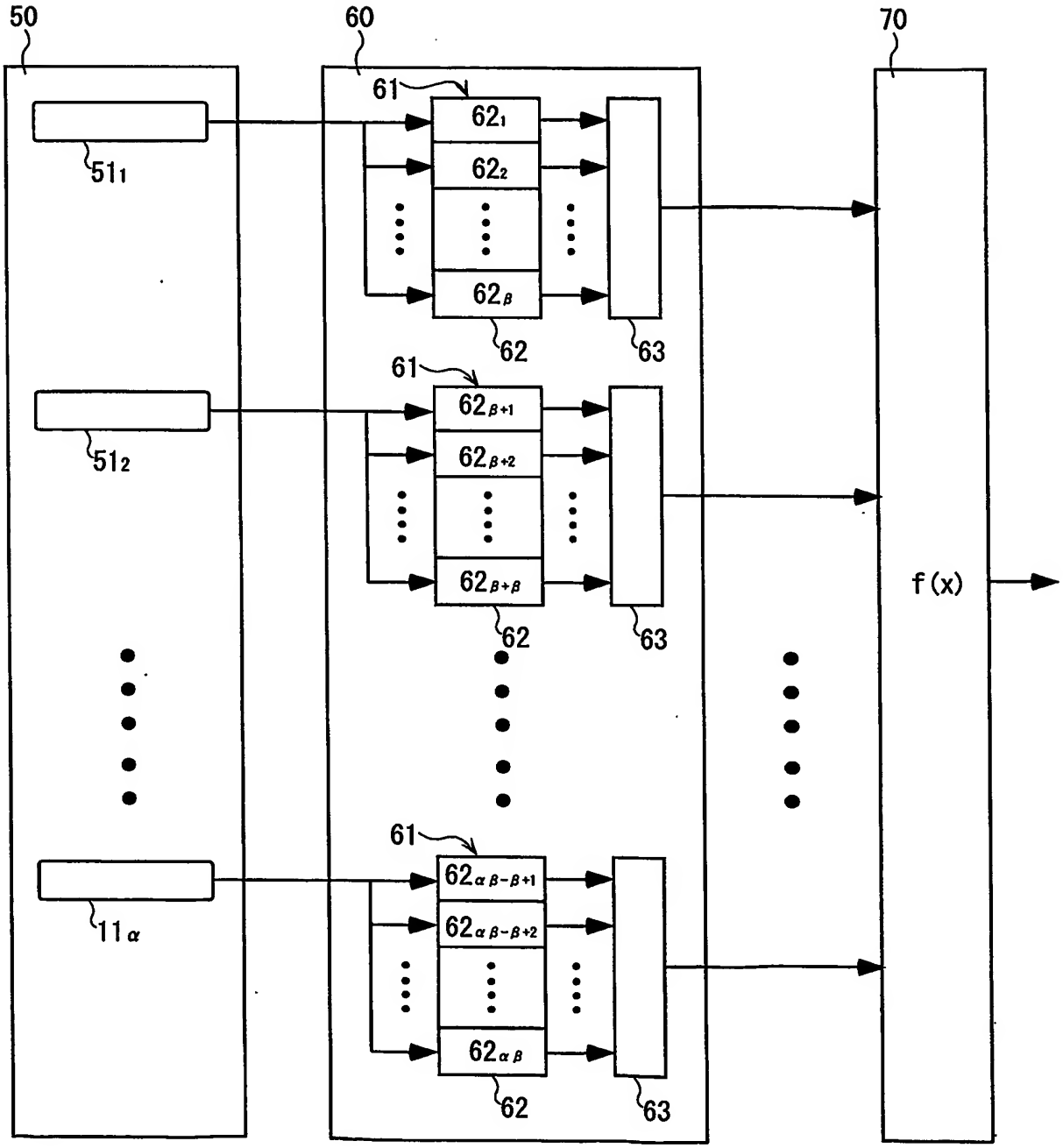
第２図

LSFR1	x^131+x^8+x^3+x^2+1
LSFR2	x^137+x^21+1
LSFR3	x^139+x^8+x^5+x^3+1
LSFR4	x^149+x^10+x^9+x^7+1
LSFR5	x^151+x^3+1
LSFR6	x^157+x^6+x^5+x^2+1
LSFR7	x^163+x^7+x^6+x^3+1
LSFR8	x^167+x^6+1

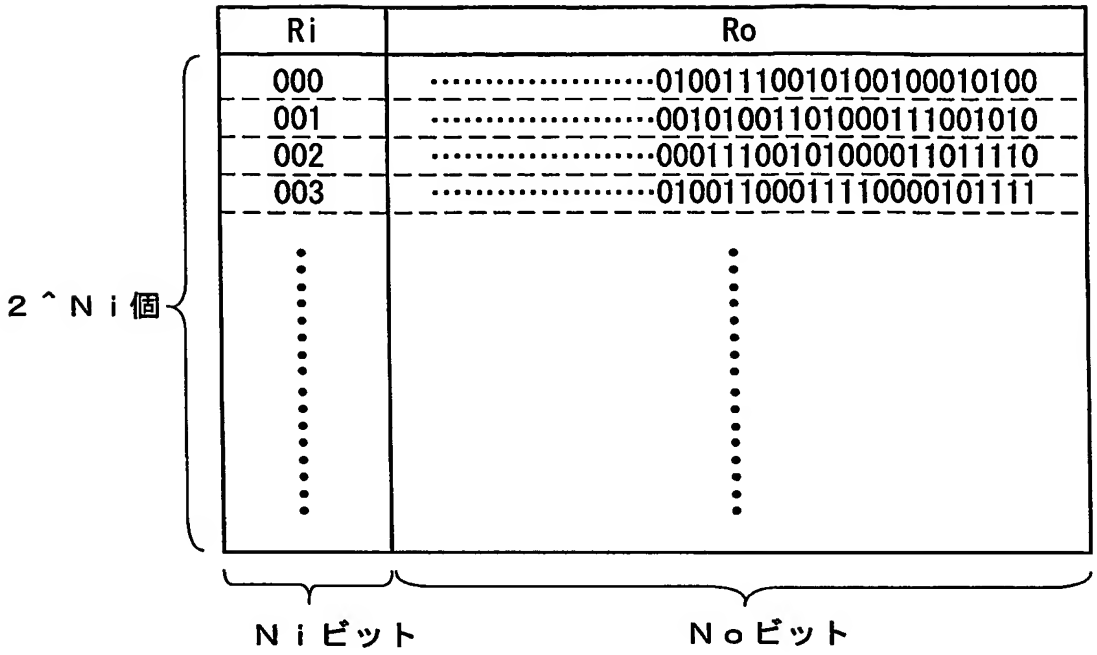
第３図


Start
初期値設定
S1
導出値s算出
S2
2ms算出
S3
2ms個のﾋﾞｯﾄ列を出力
S4
新ﾋﾞｯﾄ列生成
S5
LFSR再構成
S6
疑似乱数発生
S7
非線形変換
S8
End

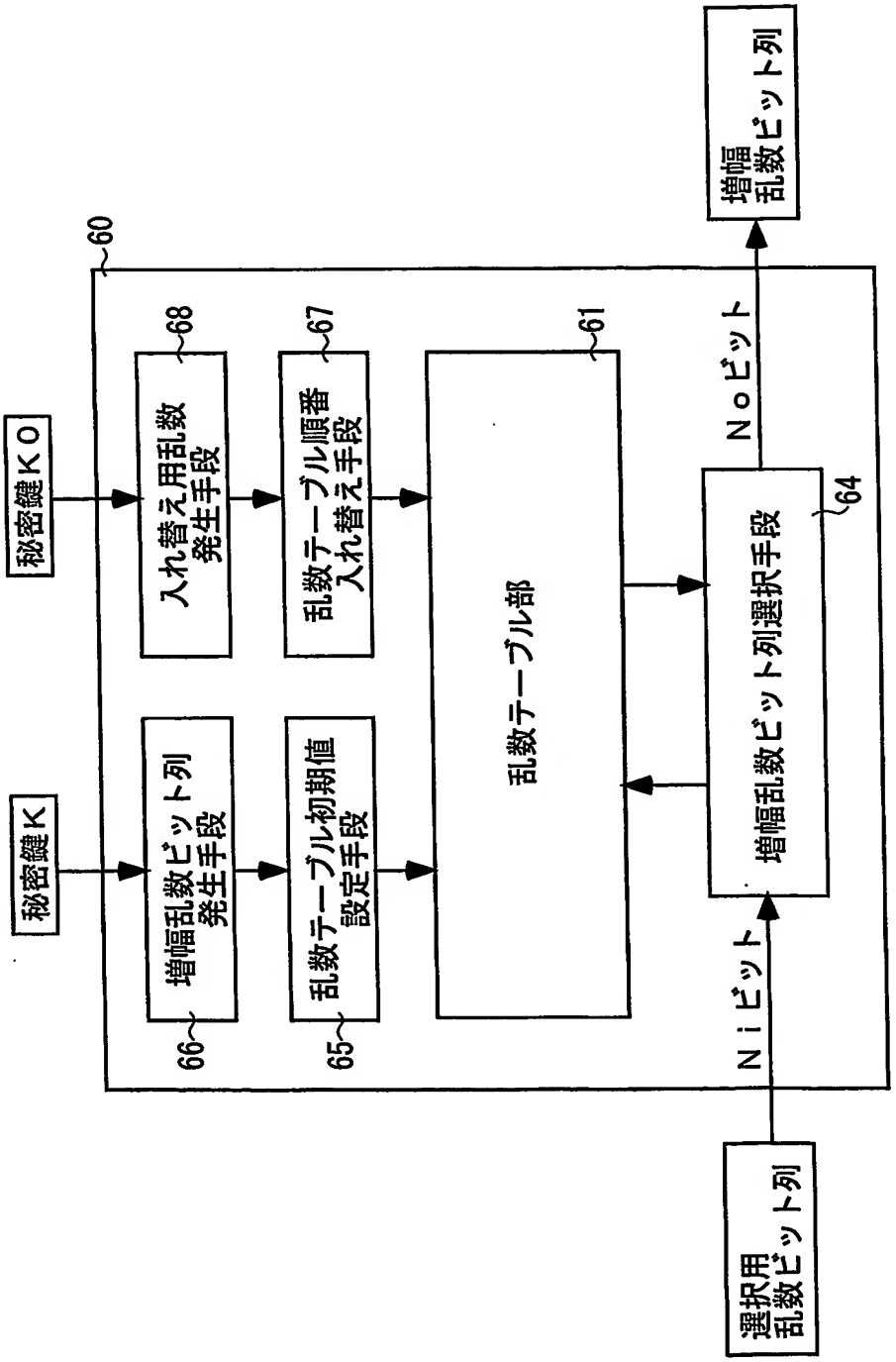
第４図


50
60
70
61
62_1
62_2
62_β
62
63
51_1
51_2
$62_{\beta+1}$
$62_{\beta+2}$
$62_{\beta+\beta}$
11_α
$62_{\alpha\beta-\beta+1}$
$62_{\alpha\beta-\beta+2}$
$62_{\alpha\beta}$
f(x)

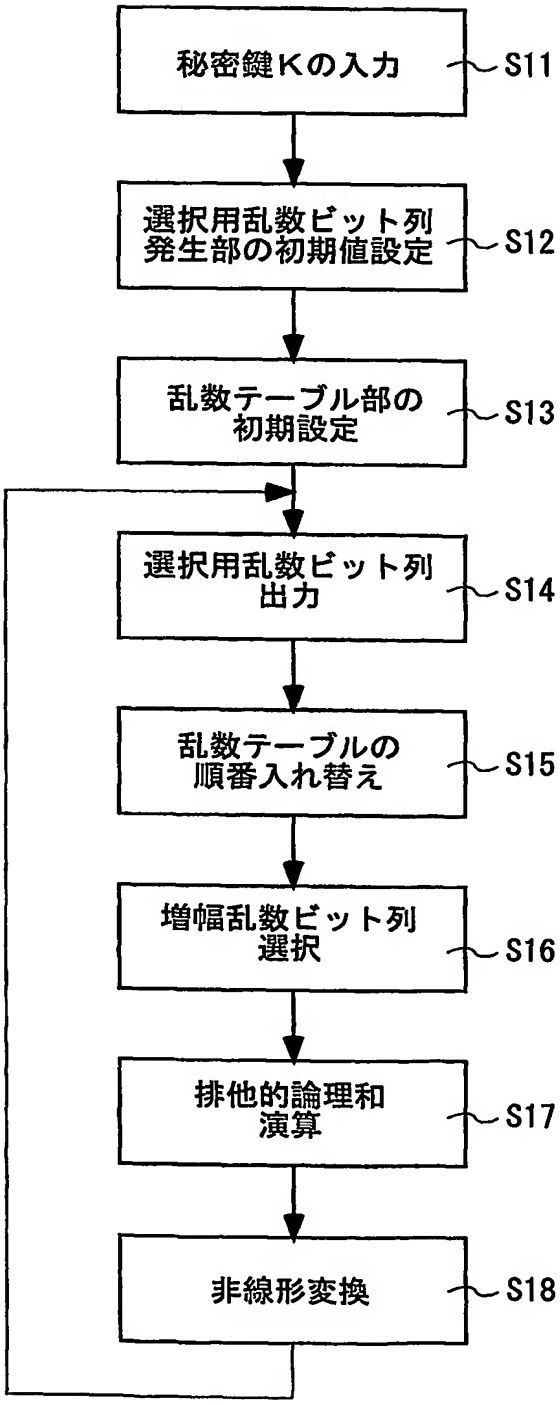
第５図



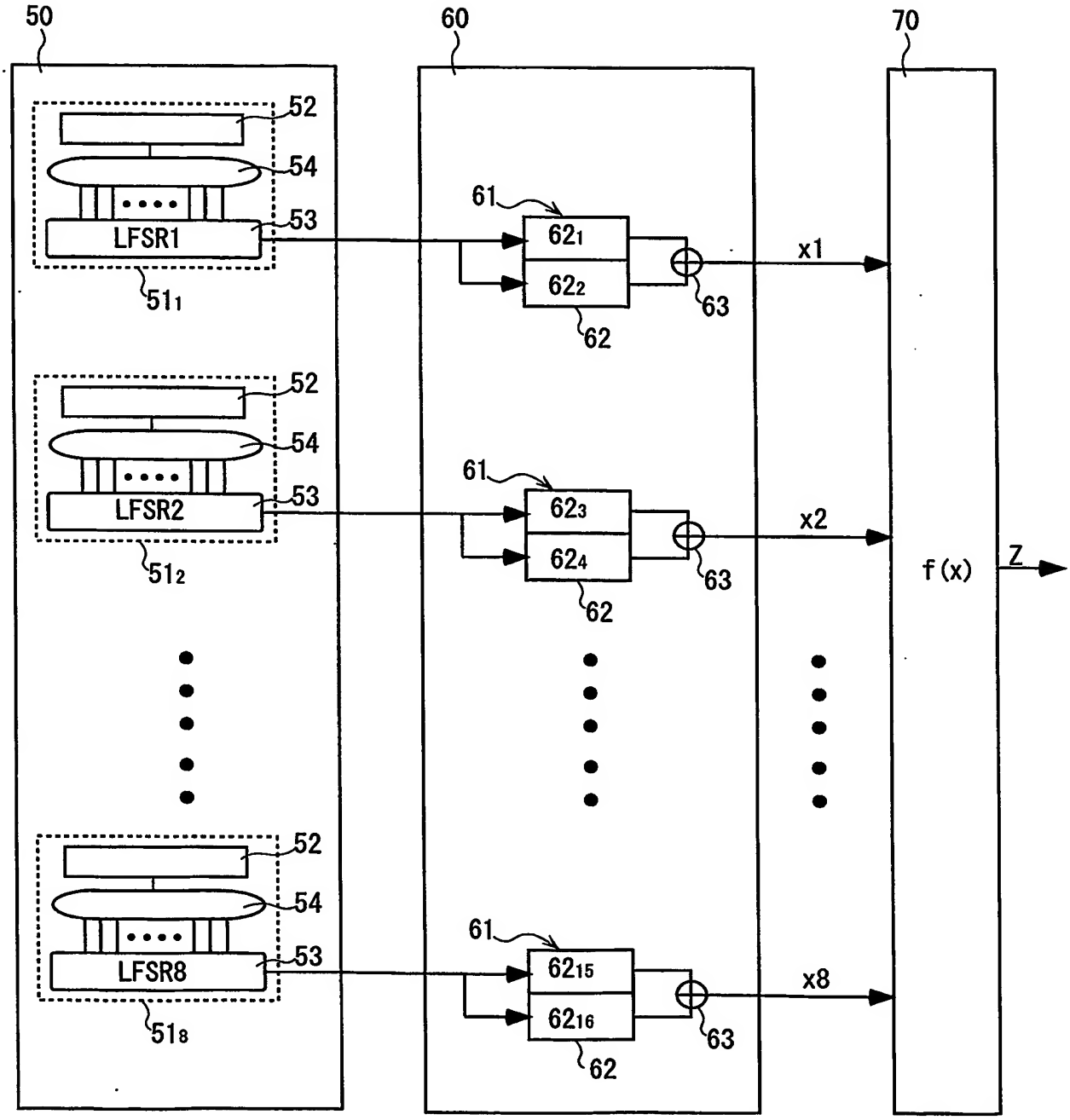
第６図


60
秘密鍵K0
秘密鍵K
入れ替え用乱数発生手段
68
乱数テーブル順番入れ替え手段
67
乱数テーブル部
61
増幅乱数ビット列発生手段
66
乱数テーブル初期値設定手段
65
増幅乱数ビット列選択手段
64
Noビット
Niビット
増幅乱数ビット列
選択用乱数ビット列

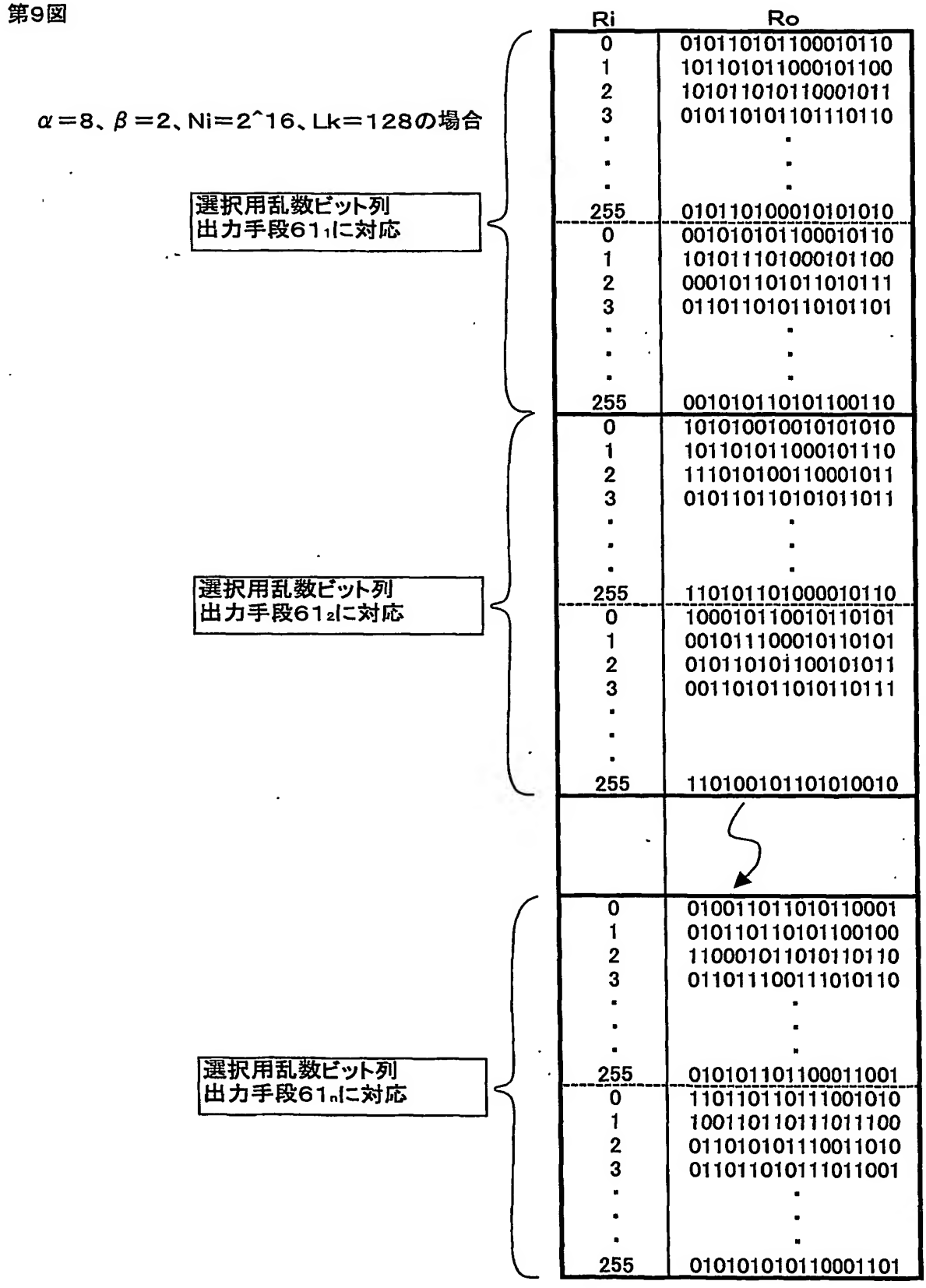
第７図


秘密鍵Kの入力
S11
選択用乱数ビット列発生部の初期値設定
S12
乱数テーブル部の初期設定
S13
選択用乱数ビット列出力
S14
乱数テーブルの順番入れ替え
S15
増幅乱数ビット列選択
S16
排他的論理和演算
S17
非線形変換
S18

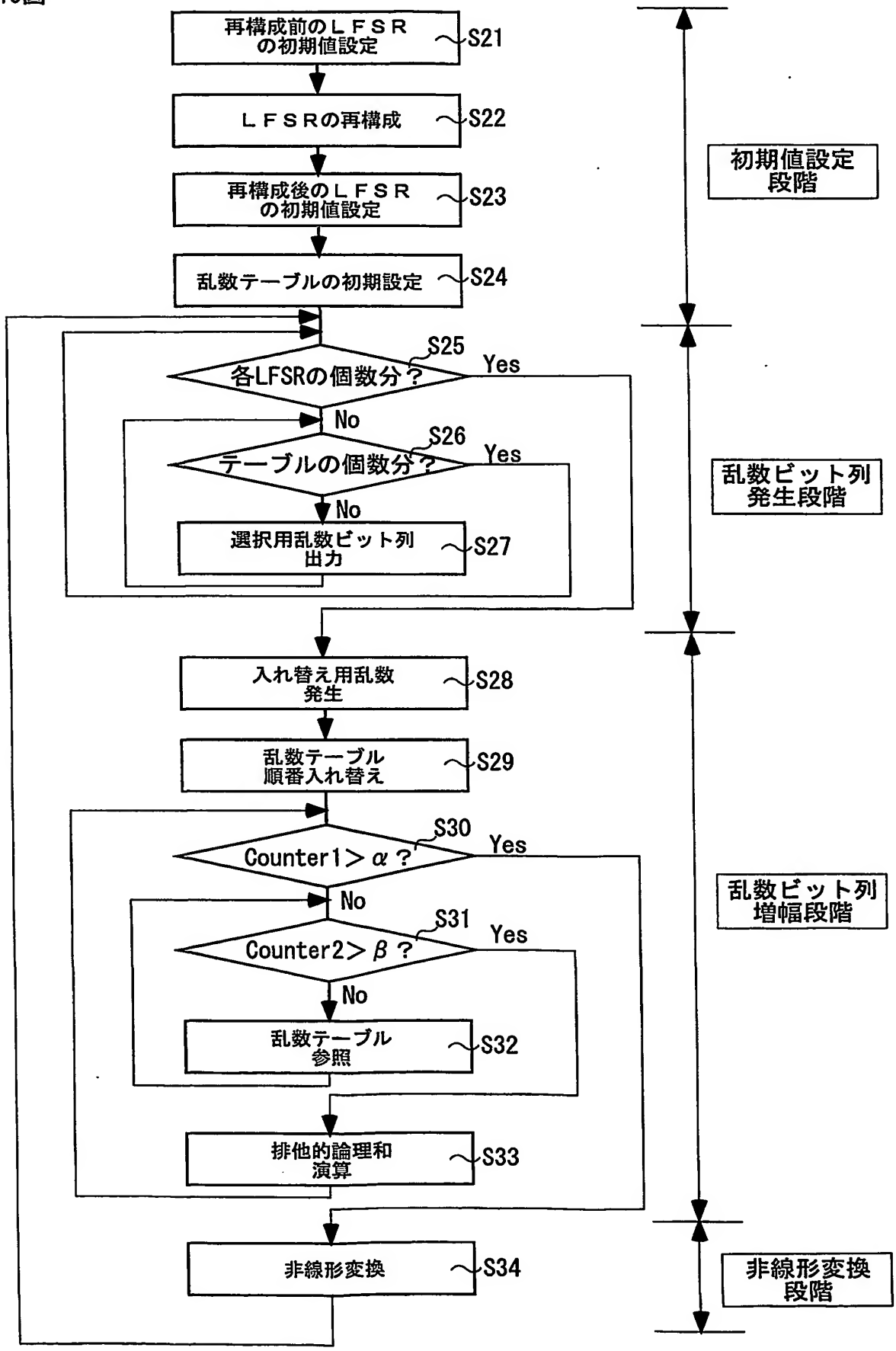
第８図


50
60
70
52
54
53
LFSR1
51₁
61
62₁
62₂
62
63
x1
LFSR2
51₂
62₃
62₄
x2
f(x)
Z
LFSR8
51₈
62₁₅
62₁₆
x8

第9図


α＝8、β＝2、Ni＝2^16、Lk＝128の場合
選択用乱数ビット列
出力手段61₁に対応
選択用乱数ビット列
出力手段61₂に対応
選択用乱数ビット列
出力手段61ₙに対応
Ri
Ro
0 01011010110001011O
1 10110101100010110
2 10101101011000101l
3 01011010110111011O
255 01011010001010101O
0 00101010110001011O
1 10101110100010110O
2 00010110101101011l
3 01101101011010110l
255 00101011010110011O
0 10101001001010101O
1 10110101100010111O
2 11101010011000101l
3 01011011010101101l
255 11010110100001011O
0 10001011001011010l
1 00101110001011010l
2 01011010110010101l
3 00110101101011011l
255 11010010110101001O
0 01001101101011000l
1 01011011010110010O
2 11000101101011011O
3 01101110011101011O
255 01010110110001100l
0 11011011011100101O
1 10011011011101110O
2 01101010111001101O
3 01101101011101100l
255 01010101011000110l

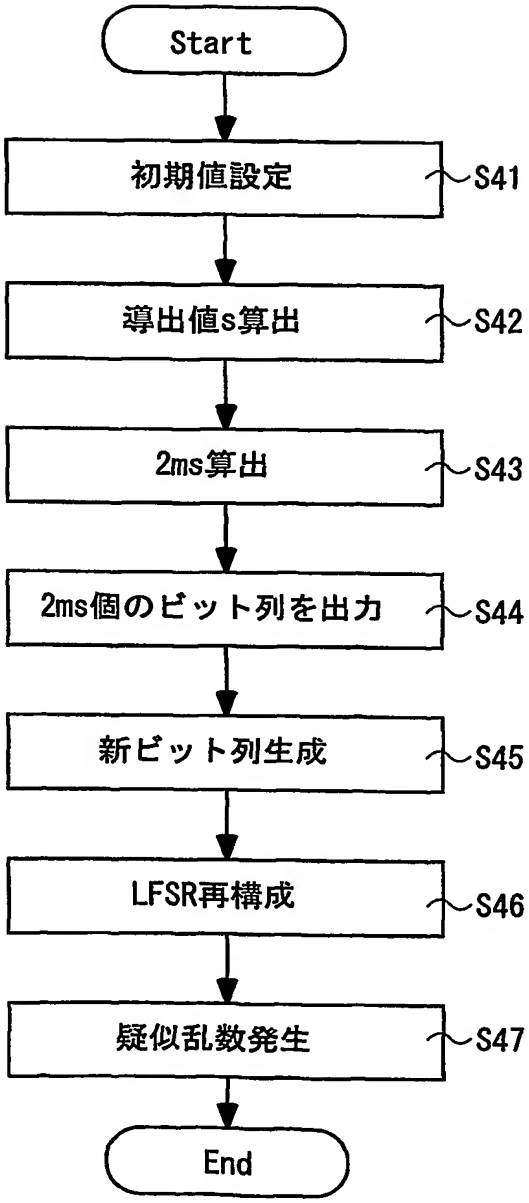
第10図


再構成前のＬＦＳＲの初期値設定
S21
ＬＦＳＲの再構成
S22
再構成後のＬＦＳＲの初期値設定
S23
乱数テーブルの初期設定
S24
初期値設定段階
S25
各LFSRの個数分？
Yes
No
S26
テーブルの個数分？
Yes
No
選択用乱数ビット列出力
S27
乱数ビット列発生段階
入れ替え用乱数発生
S28
乱数テーブル順番入れ替え
S29
S30
Counter1＞α？
Yes
No
S31
Counter2＞β？
Yes
No
乱数テーブル参照
S32
排他的論理和演算
S33
乱数ビット列増幅段階
非線形変換
S34
非線形変換段階

第11図

LFSR1	I1(x)=x^129+x^80+x^8+x+1
LFSR2	I2(x)=x^131+x^80+x^16+x+1
LFSR3	I3(x)=x^133+x^16+x^8+x^2+1
LFSR4	I4(x)=x^137+x^20+x^12+x+1
LFSR5	I5(x)=x^139+x^80+x^12+x+1
LFSR6	I6(x)=x^143+x^56+x^12+x+1
LFSR7	I7(x)=x^149+x^84+x^8+x^2+1
LFSR8	I8(x)=x^151+x^60+x^8+x+1

第12図


Start
初期値設定
S41
導出値s算出
S42
2ms算出
S43
2ms個のビット列を出力
S44
新ビット列生成
S45
LFSR再構成
S46
疑似乱数発生
S47
End

第13図

表1:スループット計測結果

LFSR(151段)の平均スループット	5.203Mbits/sec
RC4の平均スループット	141.6Mbits/sec
入れ替え処理の平均スループット	69.99Mbits/sec
乱数テーブルの平均スループット	7.923Gbits/sec
非線形関数の平均スループット	119.2Mbits/sec
疑似乱数発生器全体のスループット	116.4Mbits/sec
従来型の平均スループット	0.680Mbits/sec

第14図

表2:NISTのパラメータ

Data Length	1,000,000
Block Frequency Test Block Length	100
Non Overlapping Template Test Block Length	10
Overlapping Template Test Block Length	10
Universal Test Block Length	7
Universal Test Number Of Initialization Steps	1,280
Approximate Entropy Test Block Length	2
Serial Test Block Length	2
Linear Complexity Test Subsequence Length	500
bitstreams should be generated	10

第15図

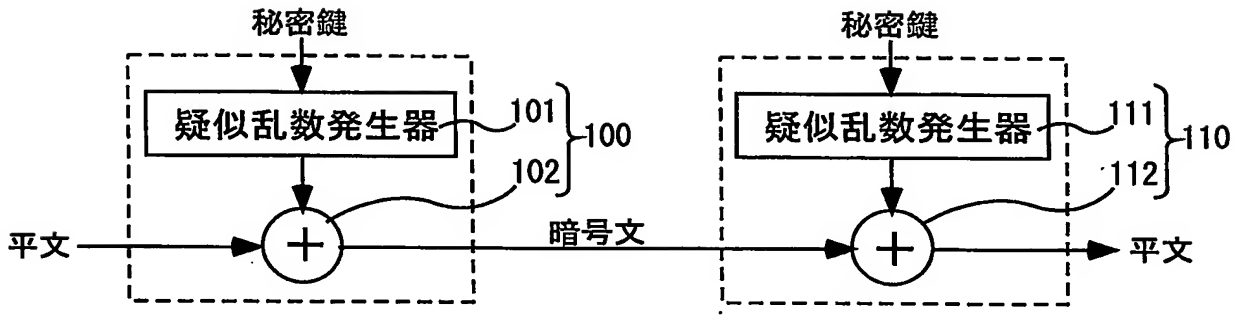
テスト項目	P-value
Frequency	0.987896
Block-Frequency	0.678686
Cusum	0.366918
Cusum	0.213309
Runs	0.383827
Long-Run	0.595549
Rank	0.798139
FFT	0.021999
Aperiodic-Template	0.867692
Aperiodic-Template	0.574903
Aperiodic-Template	0.779188
Aperiodic-Template	0.350485
Aperiodic-Template	0.798139
Aperiodic-Template	0.494392
Aperiodic-Template	0.867692
Aperiodic-Template	0.085587
Aperiodic-Template	0.474986
Aperiodic-Template	0.996335
Aperiodic-Template	0.249284
Aperiodic-Template	0.153763
Aperiodic-Template	0.514124
Aperiodic-Template	0.657933
Aperiodic-Template	0.595549
Aperiodic-Template	0.719747
Aperiodic-Template	0.996335
Aperiodic-Template	0.657933
Aperiodic-Template	0.759756
Aperiodic-Template	0.834308
Aperiodic-Template	0.851383
Aperiodic-Template	0.657933
Aperiodic-Template	0.494392
Aperiodic-Template	0.779188
Aperiodic-Template	0.883171
Aperiodic-Template	0.798139
Aperiodic-Template	0.719747
Aperiodic-Template	0.964295
Aperiodic-Template	0.401199
Aperiodic-Template	0.12962
Aperiodic-Template	0.739918
Aperiodic-Template	0.010555
Aperiodic-Template	0.202268
Aperiodic-Template	0.289667
Aperiodic-Template	0.897763
Aperiodic-Template	0.719747
Aperiodic-Template	0.494392
Aperiodic-Template	0.019188
Aperiodic-Template	0.066882
Aperiodic-Template	0.574903
Aperiodic-Template	0.699313
Aperiodic-Template	0.978072
Aperiodic-Template	0.153763
Aperiodic-Template	0.419021
Aperiodic-Template	0.851383
Aperiodic-Template	0.55442
Aperiodic-Template	0.897763
Aperiodic-Template	0.213309
Aperiodic-Template	0.319084

テスト項目	P-value
Aperiodic-Template	0.657933
Aperiodic-Template	0.494392
Aperiodic-Template	0.987896
Aperiodic-Template	0.419021
Aperiodic-Template	0.350485
Aperiodic-Template	0.616305
Aperiodic-Template	0.935716
Aperiodic-Template	0.897763
Aperiodic-Template	0.897763
Aperiodic-Template	0.171867
Aperiodic-Template	0.739918
Aperiodic-Template	0.897763
Aperiodic-Template	0.224821
Aperiodic-Template	0.851383
Aperiodic-Template	0.419021
Aperiodic-Template	0.319084
Aperiodic-Template	0.401199
Aperiodic-Template	0.616305
Aperiodic-Template	0.911413
Aperiodic-Template	0.897763
Aperiodic-Template	0.897763
Aperiodic-Template	0.897763
Aperiodic-Template	0.080519
Aperiodic-Template	0.867692
Aperiodic-Template	0.115387
Aperiodic-Template	0.275709
Aperiodic-Template	0.779188
Aperiodic-Template	0.202268
Aperiodic-Template	0.319084
Aperiodic-Template	0.637119
Aperiodic-Template	0.739918
Aperiodic-Template	0.224821
Aperiodic-Template	0.514124
Aperiodic-Template	0.137282
Aperiodic-Template	0.964295
Aperiodic-Template	0.334538
Aperiodic-Template	0.678686
Aperiodic-Template	0.719747
Aperiodic-Template	0.080519
Aperiodic-Template	0.145326
Aperiodic-Template	0.319084
Aperiodic-Template	0.145326
Aperiodic-Template	0.304126
Aperiodic-Template	0.867692
Aperiodic-Template	0.719747
Aperiodic-Template	0.437274
Aperiodic-Template	0.030806
Aperiodic-Template	0.224821
Aperiodic-Template	0.514124
Aperiodic-Template	0.171867
Aperiodic-Template	0.010988
Aperiodic-Template	0.946308
Aperiodic-Template	0.162606
Aperiodic-Template	0.534146
Aperiodic-Template	0.574903
Aperiodic-Template	0.334538
Aperiodic-Template	0.699313

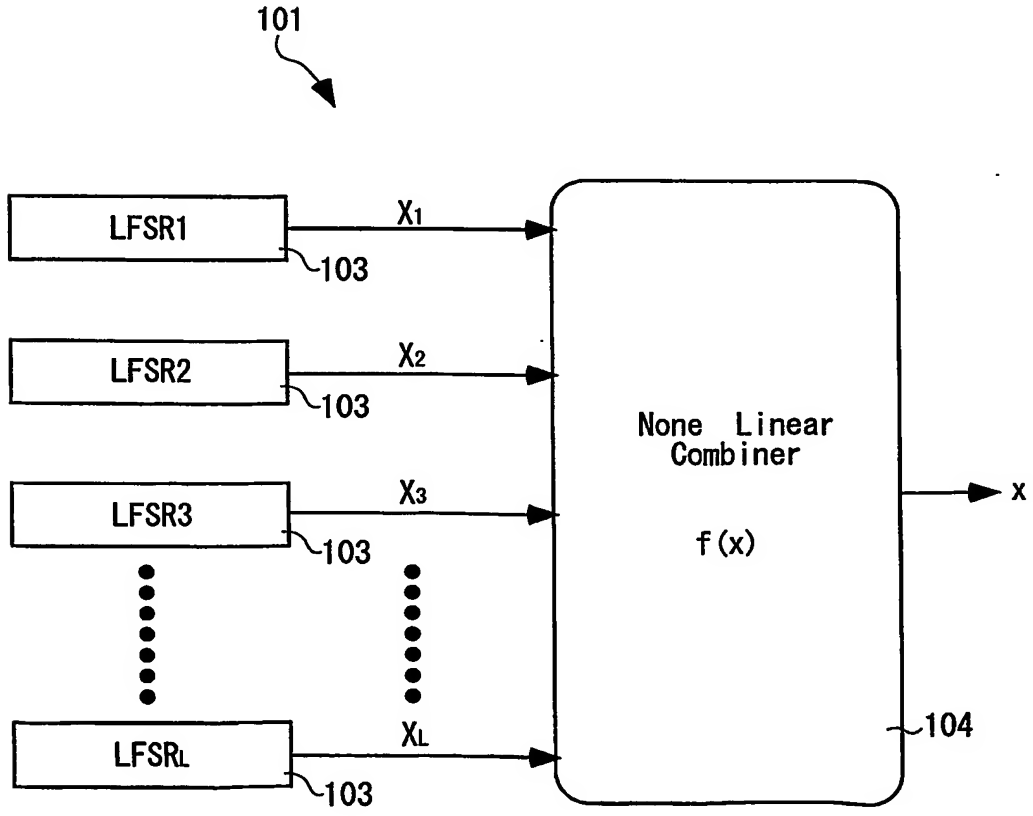
テスト項目	P-value
Aperiodic-Template	0.616305
Aperiodic-Template	0.042808
Aperiodic-Template	0.759756
Aperiodic-Template	0.779188
Aperiodic-Template	0.262249
Aperiodic-Template	0.455937
Aperiodic-Template	0.55442
Aperiodic-Template	0.383827
Aperiodic-Template	0.12962
Aperiodic-Template	0.867692
Aperiodic-Template	0.759756
Aperiodic-Template	0.637119
Aperiodic-Template	0.867692
Aperiodic-Template	0.955835
Aperiodic-Template	0.897763
Aperiodic-Template	0.996335
Aperiodic-Template	0.115387
Aperiodic-Template	0.383827
Aperiodic-Template	0.275709
Aperiodic-Template	0.55442
Aperiodic-Template	0.051942
Aperiodic-Template	0.595549
Aperiodic-Template	0.657933
Aperiodic-Template	0.637119
Aperiodic-Template	0.045675
Aperiodic-Template	0.924076
Aperiodic-Template	0.978072
Aperiodic-Template	0.739918
Aperiodic-Template	0.455937
Aperiodic-Template	0.657933
Aperiodic-Template	0.574903
Aperiodic-Template	0.304126
Aperiodic-Template	0.249284
Aperiodic-Template	0.289667
Aperiodic-Template	0.946308
Aperiodic-Template	0.010535
Aperiodic-Template	0.816537
Aperiodic-Template	0.739918
Aperiodic-Template	0.350485
Aperiodic-Template	0.798139
Aperiodic-Template	0.455937
Aperiodic-Template	0.145326
Periodic-Template	0.657933
Universal	0.383827
Apen	0.534146
Random-Excursion	0.534146
Random-Excursion	0.213309
Random-Excursion	0.534146
Random-Excursion	0.035174
Random-Excursion	0.534146
Random-Excursion	0.534146
Random-Excursion	0.010879
Random-Excursion	0.213309
Random-Excursion-V	0.534146
Random-Excursion-V	0.739918
Random-Excursion-V	0.213309
Random-Excursion-V	0.911413

テスト項目	P-value
Random-Excursion-V	0.035174
Random-Excursion-V	0.911413
Random-Excursion-V	0.350485
Random-Excursion-V	0.534146
Random-Excursion-V	0.350485
Random-Excursion-V	0.213309
Random-Excursion-V	0.213309
Random-Excursion-V	0.534146
Random-Excursion-V	0.534146
Random-Excursion-V	0.122325
Random-Excursion-V	0.213309
Random-Excursion-V	0.066882
Random-Excursion-V	0.035174
Serial	0.350485
Serial	0.289667
Lempel-Ziv	0.383827
Linear-Complexity	0.090936

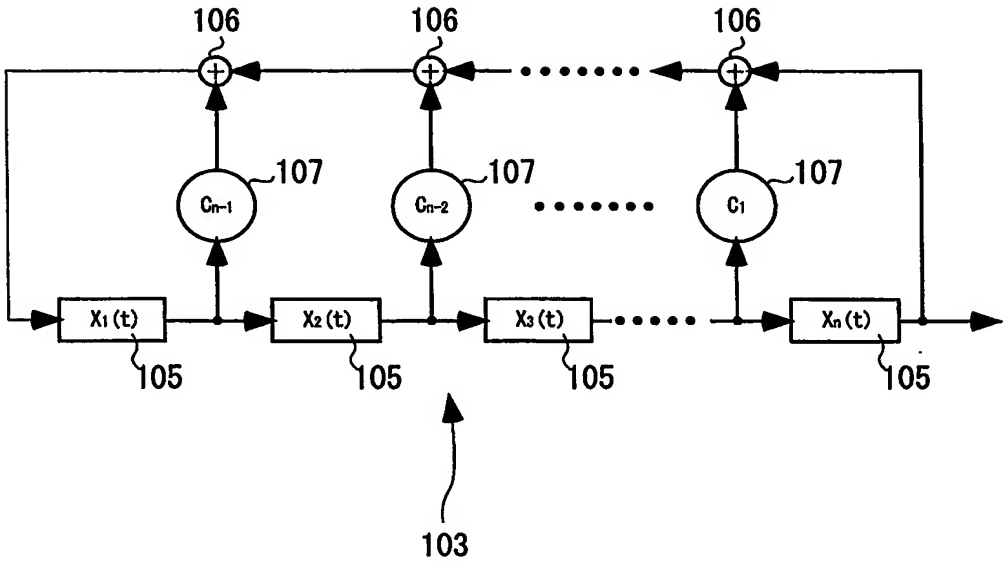
第16図


秘密鍵
疑似乱数発生器
101
100
102
平文
暗号文
秘密鍵
疑似乱数発生器
111
110
112
平文

第17図


101
LFSR1
X1
103
LFSR2
X2
103
LFSR3
X3
103
LFSRL
XL
103
None Linear Combiner
f(x)
x
104

第18図


106
106
106
107
107
107
Cn-1
Cn-2
C1
X1(t)
X2(t)
X3(t)
Xn(t)
105
105
105
105
103

第Ⅷ欄（v）　不利にならない開示又は新規性喪失の例外に関する申立て

申立ては実施細則第215号に規定する標準文言を使用して作成しなければならない。第Ⅷ欄と同欄(i)～(v)の備考の総論部分、及び本頁に特有の事項について第Ⅷ欄(v)の備考を参照。この欄を使用しないときは、この用紙を願書に含めないこと。

不利にならない開示又は新規性喪失の例外に関する申立て（規則4.17(v)及び51の2.1(a)(v)）

本国際出願［PCT/JP03/08794］に関し、

小林　朗、森井　昌克、白石　善明は、本国際出願の請求項に記載された対象が以下のように開示されたことを申し立てる。

(i)　開示の種類：刊行物

(ii)　開示の日付：2003年1月26日

(iii)　開示の名称：2003年　暗号と情報セキュリティシンポジウム予稿集Vol.II

(iv)　開示の場所：静岡県浜松市板屋町　アクトシティ浜松　コングレスセンター

(v)　本申立ては、すべての指定国のためになされたものである。

☐ この申立ての続葉として「第Ⅷ欄(v)の続き」がある

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP03/08794

A. CLASSIFICATION OF SUBJECT MATTER
Int.Cl[7] G09C1/00, G06F7/58

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
Int.Cl[7] G09C1/00, G06F7/58

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho	1922-1996	Toroku Jitsuyo Shinan Koho	1994-2003
Kokai Jitsuyo Shinan Koho	1971-2003	Jitsuyo Shinan Toroku Koho	1996-2003

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X Y	JP 10-240500 A (Toshiba Corp.), 11 September, 1998 (11.09.98), Figs. 3 to 12 (Family: none)	7-10,14-17 11-13,18-20
X	JP 9-179726 A (NEC Corp.), 11 July, 1997 (11.07.97), Full text & EP 782069 A	7,9,14,16
X	JP 62-144243 A (NEC Corp.), 27 June, 1987 (27.06.87), Full text (Family: none)	7,9,14,16

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier document but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
26 August, 2003 (26.08.03)	09 September, 2003 (09.09.03)
Name and mailing address of the ISA/ Japanese Patent Office Facsimile No.	Authorized officer Telephone No.

Form PCT/ISA/210 (second sheet) (July 1998)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP03/08794

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	JP 4-115616 A (Toshiba Corp.), 16 April, 1992 (16.04.92), Full text (Family: none)	11-13,18-20
A	JP 7-36672 A (Canon Inc.), 07 February, 1995 (07.02.95), Full text & EP 635956 A	1-20
A	JP 7-104976 A (NEC Corp.), 21 April, 1995 (21.04.95), Full text & US 5566099 A	1-20
A	JP 61-246787 A (Fujitsu Ltd.), 04 November, 1986 (04.11.86), Full text (Family: none)	7-20

国際調査[redacted]

国際出願番号 [redacted]CT／JP03／08794

A. 発明の属する分野の分類（国際特許分類（IPC））

Int. Cl⁷ G09C 1／00 , G06F 7／58

B. 調査を行った分野

調査を行った最小限資料（国際特許分類（IPC））

Int. Cl⁷ G09C 1／00 , G06F 7／58

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	1922－1996年
日本国公開実用新案公報	1971－2003年
日本国登録実用新案公報	1994－2003年
日本国実用新案登録公報	1996－2003年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

C. 関連すると認められる文献

引用文献のカテゴリー＊	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求の範囲の番号
X Y	JP 10－240500 A（株式会社東芝） 1998.09.11, 第3図－第12図（ファミリーなし）	7－10, 14－17 11－13, 18－20
X	JP 9－179726 A（日本電気株式会社） 1997.07.11, 全文 & EP 782069 A	7, 9, 14, 16

☒ C欄の続きにも文献が列挙されている。　☐ パテントファミリーに関する別紙を参照。

＊ 引用文献のカテゴリー

「A」特に関連のある文献ではなく、一般的技術水準を示すもの

「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの

「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）

「O」口頭による開示、使用、展示等に言及する文献

「P」国際出願日前で、かつ優先権の主張の基礎となる出願の日の後に公表された文献

「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの

「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの

「Y」特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの

「&」同一パテントファミリー文献

国際調査を完了した日	国際調査報告の発送日
26.08.03	09.09.03

国際調査機関の名称及びあて先	特許庁審査官（権限のある職員）	
日本国特許庁（ISA／JP） 郵便番号100－8915 東京都千代田区霞が関三丁目4番3号	石田 信行	5M 9469
	電話番号 03－3581－1101 内線 3598	

様式PCT／ISA／210（第2ページ）（1998年7月）

Ｃ（続き）．　関連すると認められる文献

引用文献のカテゴリー＊	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求の範囲の番号
Ｘ	ＪＰ　６２－１４４２４３　Ａ（日本電気株式会社） １９８７．０６．２７，全文（ファミリーなし）	７，９， １４，１６
Ｙ	ＪＰ　４－１１５６１６　Ａ（株式会社東芝） １９９２．０４．１６，全文（ファミリーなし）	１１－１３， １８－２０
Ａ	ＪＰ　７－３６６７２　Ａ（キヤノン株式会社） １９９５．０２．０７，全文　＆　ＥＰ　６３５９５６　Ａ	１－２０
Ａ	ＪＰ　７－１０４９７６　Ａ（日本電気株式会社） １９９５．０４．２１，全文　＆　ＵＳ　５５６６０９９　Ａ	１－２０
Ａ	ＪＰ　６１－２４６７８７　Ａ（富士通株式会社） １９８６．１１．０４，全文（ファミリーなし）	７－２０